

JNC

REPORT DOCUMENTATION PAGE

AD-A199 993

CT 0 6 1088
JUL 88

1b. RESTRICTIVE MARKINGS

3. DISTRIBUTION/AVAILABILITY OF REPORT

Unrestricted

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
447222-3

5. MONITORING ORGANIZATION REPORT NUMBER(S)

AFOSR-TR- 88-0988

6a. NAME OF PERFORMING ORGANIZATION
Dept. of E.E.
The City College of New York6b. OFFICE SYMBOL
(If applicable)7a. NAME OF MONITORING ORGANIZATION
Dr. Lee Giles
AFOSR/NE6c. ADDRESS (City, State, and ZIP Code)
138th St. and Convent Ave.
New York, N.Y. 100317b. ADDRESS (City, State, and ZIP Code)
Building 410
Bolling AFB, D.C. 20332-64488a. NAME OF FUNDING/SPONSORING
ORGANIZATION8b. OFFICE SYMBOL
(If applicable)9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
AFOSR-85-0212

8c. ADDRESS (City, State, and ZIP Code)

10. SOURCE OF FUNDING NUMBERS

PROGRAM
ELEMENT NO.PROJECT
NO.TASK
NO.WORK UNIT
ACCESSION NO.

11. TITLE (Include Security Classification)

Optical Acquisition, Image and Data Compression UNC

12. PERSONAL AUTHOR(S)
Eichmann, George13a. TYPE OF REPORT
Final13b. TIME COVERED
FROM 6/87 TO 7/8814. DATE OF REPORT (Year, Month, Day)
July 30, 198815. PAGE COUNT
97

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD

GROUP

SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

Optical Information Processing

Optical Computing, Associative Memory

Optical Artificial Intelligence

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

→ A new conditional symbolic substitution rule for modified signed-digit arithmetic computation is introduced. Using this substitution rule, the numbers to be added or subtracted are first replaced by a pair of new equivalent strings, which in a second step are then subject to another substitution to generate both the addition or subtraction result and its complement. For an optical implementation, a holographic content-addressable memory is used. Correspondingly, the input encoding, the logic reduction, and the optical processing techniques are described.

An optical isochronous array processing method is proposed. An optical isochronous array processor (OIAP) is a local regularly interconnected processing network that employs an array of identical optical processing elements. In an OIAP, incoming isochronous data are parallel processed in a fashion much like a propagating electromagnetic wavefront. For the various applications, the OIAP processing elements and their interconnections can be different. In this paper, various all-optical OIAP elements are considered. Applications ranging from → 0062

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT

☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION

22a. NAME OF RESPONSIBLE INDIVIDUAL

22b. TELEPHONE (Include Area Code)

22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted.

All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

UNC

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

88 10 5 212

optical binary number multiplication preprocessing to optical matrix algebra as well as to optical residue arithmetic are presented.

After a short introduction, a brief overview of the research papers contained in this special is presented.

There are many pattern recognition problems where the pattern's structural information is important. In these problems, a syntactic method of pattern recognition is of value. In this paper, both parallel syntactic pattern recognition algorithm and optical architecture implementation approaches are described. In particular, the applications of syntactic pattern recognition algorithm to shape classification are illustrated. A number of parallel optical syntactic pattern coding methods, a structural matched filter and associative memory filter, and an optical symbolic substitution syntactic parser are discussed.

The long-standing problem of the superresolving reconstruction (restoration) of an object of known finite spatial extent from a noisy linearly degraded image is considered. The resolution of two-point sources (objects) spaced less than one Rayleigh distance apart is an ill-posed problem. To determine a superresolving inverse of an ill-conditioned linear degradation operator with a known set of input/output training signals, a linear associative memory (LAM) technique is employed. By limiting the set of reconstructable signals, an exceptionally robust inverse filter has been obtained. This filter is based on a new constrained LAM matrix operator technique. Superresolving restoration of 1-D and 2-D two-point sources as well as some typical edge-type signals in the presence of considerable measurement noise is demonstrated.

Texture is one of the important image characteristics and is used to identify objects or regions of interest. The problem of texture classification has been widely studied. Texture classification techniques are either statistical or structural. Some statistical texture classification approaches use Fourier power-spectrum features, while others are based on first- and second-order statistics of gray level differences. Periodic textures that consist of mostly straight lines are of particular interest. In this paper, a new structural approach based on the Hough method of line detection is introduced. This classification is based on the relative orientation and location of the lines within the texture. With proper normalization, the classification is independent of geometrical transformation such as rotation, translation and/or scaling. Experimental results will also be presented.

The use of the optical phase-conjugation (OPC) process for optical residue computation is proposed. By using an OPC-based parallel switching array, various optical position-coded residue-mapping units for carry-free addition, subtraction, and multiplication operations are described. Experimental results obtained with a picosecond mode-locked Nd^{3+} :YAG laser are presented to support the proposal.

Pyramidal processing is a form of multiresolution image processing in which the image is decomposed into a sequence of images at different resolutions. Pyramidal processing aims to extract and interpret significant features of an image at different resolutions. Digital pyramidal image processing, because of the large number of convolution-type operations, is time consuming. On the other hand, optical pyramidal processors, described here, are preferable in real-time image-understanding applications because of their ease in performing convolution operations. Preliminary experimental results for optical Gaussian and Laplacian pyramidal image processing are presented.

A parallel optical binary multiplication scheme is proposed in which parallel convolution preprocessing is performed using a parallel-input optical outer-product processor together with a one dimensional either space or time integrator. Using a theta-modulation based optical A/D converter and a carry look-ahead adder array, the resulting mixed-binary partial product can be reduced to the final binary multiplication result.

Median filters (MF) are used both to filter 'salt and pepper' noise from signals and images and in other signal processing applications. In this paper, an extension of the MF, the vector median filter (VMF), is introduced. As opposed to the MF, the VMF outputs for each window location a number of data elements. By adjusting the VMF parameters, the MF is obtained as a VMF special case. Just like the MF, the VMF filters impulses while simultaneously preserving step changes in a signal. The VMF's principal advantage is that it reduces

TABLE OF CONTENTS

AFOSR-TN- 88-0280

1. Conditional symbolic modified signed-digit arithmetic using optical content-addressable memory logic elements	1
2. Digital optical isochronous array processing	7
3. Optical artificial intelligence and symbolic computing: an introduction	15
4. Parallel optical syntactic pattern recognizers	17
5. Superresolving signal and image restoration using a linear associative memory	24
6. Topologically invariant texture descriptors	32
7. Demonstration of a picosecond optical-phase-conjugation-based residue-arithmetic computation	47
8. Parallel optical pyramidal image processing	50
9. Fast parallel optical digital multiplication	53
10. Vector median filters	59
11. Real-time optical line detection	74
12. Compact optical generalized perfect shuffle	77
13. Parallel optical logic using optical phase conjugation	80
14. An AND operation-based optical symbolic substitution pattern recognizer	83
15. Multistable Fabry-Perot resonator with an active Sagnac interferometer as its retro-reflector	88
16. Hough-transform-based circle detection using an array of multimode optical fibers	94

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

Conditional symbolic modified signed-digit arithmetic using optical content-addressable memory logic elements

Yao Li and George Eichmann

A new conditional symbolic substitution rule for modified signed-digit arithmetic computation is introduced. Using this substitution rule, the numbers to be added or subtracted are first replaced by a pair of new equivalent strings, which in a second step are then subject to another substitution to generate both the addition or subtraction result and its complement. For an optical implementation, a holographic content-addressable memory is used. Correspondingly, the input encoding, the logic reduction, and the optical processing techniques are described.

1. Introduction

Two of the fundamental arithmetic operations in digital computing are addition and multiplication. A binary electronic digital computer addition algorithm is based on a carry propagation that severely limits its computing speed. The conventional binary multiplication, because it uses a number of shift and add operations, further reduces its speed. To increase its speed, nonbinary number representations and their corresponding addition and multiplication algorithms have been investigated. Multiple-valued fixed radix number representation, with their higher logic or number density, can lead to a processing speed increase by propagating more information through each interconnection.¹⁻³ Using a residue number system⁴ both carry-free addition and multiplication operations can be performed. The residue method decomposes an arithmetic operation into a number of independent suboperations to be performed by different prime modulo-based logic elements. Using a combination of residue number systems and optical parallel processing techniques, optical residue processors and the corresponding optical encoding techniques have been proposed and implemented.⁵⁻⁷ Although it is a parallel processing scheme, when large numbers need to be processed, large prime modulo logic elements, elements that are difficult to implement, must be used. Furthermore, since the different modulo subprocessors create a non-

symmetric system architecture requiring many delay elements, these architectures can also lead to a speed reduction. Thus, to gain the advantage of parallel processing while reducing the implementation complexity, other number representations and algorithms have been sought.

It has been indicated that the modified signed-digit (MSD) number representation is a candidate for fast parallel digital optical arithmetic processing operations.⁸⁻¹² The MSD, originally proposed by Avizienis⁸ and introduced to optics community by Drake, Bocker, and co-workers,¹⁰ uses a redundant binary representation that, after several parallel transfer and weight operations, leads to carry-free addition and subtraction. To synthesize optically both the transfer and weight logic functions, Drake *et al.* proposed a location-addressable memory (LAM) logic. Bocker *et al.*¹¹ generalized this method by using several symbolic substitution rules. With either method, addition (subtraction) can be performed in three (four) logic steps. Recently, Mirsalehi and Gaylord (MG)¹² proposed for the MSD addition a direct truth-table look-up content-addressable memory (CAM). Since this is a one-step method, it does increase addition speed. However, since for each output bit fifty-six Fourier holograms need to be stored, implementation difficulties, such as the practical holographic crystal multiplexibility and readout efficiency, may be encountered. In this paper, first, a new set of conditional symbolic substitution rules that uses instead of bit-wise substitution a pair of reference bits for conditional bit-wise symbolic substitutions is suggested. With this method, both addition and subtraction operations can be implemented in two logic steps. For the optical synthesis of the required MSD logic elements, a holographic CAM technique is employed. With our method, for each logic output bit only a storage of up to

The authors are with City University of New York, City College, Department of Electrical Engineering, New York, New York 10031.
Received 5 December 1986.
0003-6935/87/122328-06\$02.00/0.
© 1987 Optical Society of America.

Table I. MSD Addition Six Possible Bit-Pair Carries and Sums

+	1	0	0	0	1	0	1
carry	1	0	0	0	1	0	1
sum	0	0	0	0	1	1	1

twelve reference holograms is required. Compared to the MG approach, while the processing speed decreases by a half, the burden on the high density holographic storage is relaxed.

The paper is organized as follows: after this Introduction, in Sec.II, the MSD number representation is briefly discussed, and the new two-step addition and subtraction conditional symbolic substitution rules are given. In Sec.III, using a holographic CAM method, an optical implementation of the two-step symbolic substitution rules is presented. The corresponding truth-table logic minimization and signal encoding and recording are also described. Finally, in Sec.IV a summary of the paper is given.

II. Conditional Symbolic Modified Signed-Digit Arithmetic Operations

The MSD is a redundant radix-two number representation. Using a MSD, a number A can be represented as a string of symbols:

$$A = \sum a_i 2^i \quad (1)$$

where the digits are $a_i \in \{-1, 0, +1\}$. In this paper, only MSD integer arithmetic operations are discussed. With its balanced weighting factors, the MSD can represent both positive and negative numbers. For example, the decimal numbers $A = 11_{10}$ and $B = -11_{10}$ in the MSD system are

$$A = 11111_{\text{MSD}} \quad (2a)$$

$$B = 11111_{\text{MSD}} \quad (2b)$$

where 1 denotes a negative one. A negative number can be obtained from its positive counterpart by taking a bit-wise logic complement. For this reason, the MSD-based subtraction operation can be performed using a complement and add operation.

First, details of MSD addition operation are discussed. For a MSD bit-wise addition, the six possible bit pairs to be added (see also Table I) are (1,1), (1,1), (0,0), (1,1), (0,1) and (0,1). The first two cases generate a nonzero carry to its next higher level bit position, while the third and fourth bit pairs will, on the other hand, produce a zero carry to its left neighbor. The fourth case is particularly interesting, since it can be used to stop carry propagation. The last two bit pairs can give redundant results where either no carry or a different signed carry is generated, i.e., either

$$0 + 1 = 1 + 0 = 01_{\text{MSD}} \text{ and } 0 + \bar{1} = \bar{1} + 0 = 01_{\text{MSD}} \quad (3a)$$

or

Table II. First MSD Addition Conditional Symbolic Substitution Rule Truth Table for Rearranging Data: T_+ (W_+), the Corresponding Transfer (Weight) Operator

X_i	Y_i	$(Y_i - X_i - 1)$	T_+	W_+
11	11	10	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11	1	0
00	11	10	1	0
11	00	11	1	0
01	11	11	1	0
01	00	10	1	0
11	01	11	1	0
01	11	11	1	0
11	10	11	1	0
10	11	11		

Table III. Second MSD Addition Bit-wise Symbolic Substitution Rule Truth Table: A (C), the Sum (Complement of the Sum) Operator

		Y_i		
X_i		1	0	$\bar{1}$
	1		1	$\bar{0}$
	0	1	$\bar{1}$	0
	$\bar{1}$	0	0	1

Table IV. First MSD Subtraction Conditional Symbolic Substitution Rule Truth Table [Converting the Subtraction to Addition Operation and Rearranging the Addition Data: T_- (W_-), the Corresponding Transfer (Weight) Operator]

		Y_{i-1}		
X_{i-1}		1	0	$\bar{1}$
	1		1	$\bar{0}$
	0	1	$\bar{1}$	0
	$\bar{1}$	0	0	1

our new symbolic substitution rules is performed. In the MSD representation, this addition is

$$\begin{array}{r} 101110101011 \\ + 111100110110 \\ \hline \end{array} \quad (4)$$

After the conditional substitution, the addition of Eq. (4) becomes

$$\begin{array}{r} 111010000111\phi \\ + \phi 010010011101 \\ \hline \end{array} \quad (5)$$

where two padding zeros denoted ϕ are introduced.

Using the second substitution rule, the final result is

$$100110010011 \quad \text{for the addition result.} \quad (6a)$$

$$1100110010011 \quad \text{for the complement of addition.} \quad (6b)$$

Equation 6 is the MSD representations for the positive and negative number $\pm 2191_{10}$.

The conventional MSD subtraction operation uses a complement followed by an addition substitution rules. This method uses four logic steps¹¹ that consume both processing time and logic elements. Since with the MSD arithmetic, a single-step full-adder does not exist, this four-step method is inefficient. To increase the MSD subtraction speed, next, a new two-step conditional symbolic substitution method is suggested. Using a combination of MSD logic complement and the above-mentioned addition number rearrangement rule, the first three of the four conventional processing steps can be combined. In Table IV, where the operators T_- and W_- denote the transfer and weight functions, respectively, a conditional bit-wise subtraction substitution rule is summarized. Because with subtraction operation, the minuend and subtrahend cannot be interchanged, a 9×9 truth table is used. After this substitution rule and using the previously defined A and C operators (see Table III), the rearranged numbers are added to generate the final result and its complement. Taking the previously used numbers as a subtraction example, we have

$$\begin{array}{r} 101110101011 \\ - 111100110110 \\ \hline \end{array} \quad (7)$$

After the T_- and W_- operations, the new bit strings are

$$\begin{array}{r} 000110111001\phi \\ + \phi 010010011101 \\ \hline \end{array} \quad (8)$$

where again two padded zeros are used. Using the A and C operators, the two results are added to produce

$$001111101111 \quad \text{for the subtraction result.} \quad (9a)$$

$$001111101111 \quad \text{for the complement of subtraction.} \quad (9b)$$

Therefore, instead of the four-step substitution, the new method generates the carry-free parallel subtraction in two successive substitution steps.

In Fig. 1, using this two-step substitution algorithm for $X \pm Y$, a MSD addition/subtraction flow diagram is shown. In the first substitution step, either T_+, W_+ , or T_-, W_- operations are performed, while in the second step the logic functions A and C are used. In the second step, the two A logic gates indicated by dashed line boxes can be omitted. Thus, to perform a two N -bit word parallel addition/subtraction, $NT, NW, (N-1)A$, as well as $(N+1)C$ logic gates are required. Since it is a very regular structure, it can easily be extended to perform both addition and subtraction for any size numbers. Using additional shift operators, MSD multiplication can also be performed. With the shift operators, first, the partial products are formed. The parallel adders, forming a tree structure, are then used to add the intermediate results to yield the final product.^{9,10}

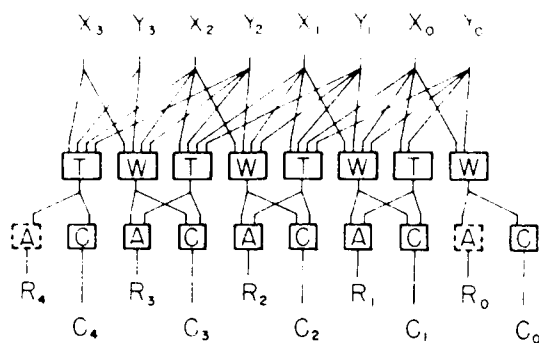


Fig. 1. Four digit MSD addition (subtraction) network. $X_i Y_i$, input strings; $T(W)$, transfer (weight) operators for addition or subtraction; $A(C)$, operators to obtain the final addition (subtraction) result R and its complement C . The operators indicated within the dashed boxes may be deleted.

III. Content-Addressable Memory MSD Logic Processing

For the MSD operation, Drake *et al.* employed a LAM method.¹⁰ To implement each of the four LAM logic functions, twenty-seven optical elements, including arrays of holograms, etalons, as well as prisms, are needed. It has been indicated that a CAM is more efficient than its LAM counterpart.¹²⁻¹⁵ With a CAM, the truth-table outputs are first classified as to their logic levels. For each output level, corresponding to different input combinations, either a sum of product or a product of sum expression is obtained. With the aid of a Karnaugh map, a reduced logic expression (a reference pattern) is generated. These patterns are then stored in an optical memory. By classifying and addressing its content rather than its truth-table location, the number of memory elements can be reduced. Using a CAM, direct binary truth-table look-up parallel addition and multiplication techniques have been proposed.¹¹ However, for long bit strings, a large truth table (even after minimization) must be constructed. To alleviate this problem, a residue arithmetic-based CAM has also been utilized.^{13,14} With a residue-based CAM, for example, for the addition of two 16-bit numbers, the truth table is dramatically reduced, i.e., from the order of billion to only a few hundred elements.¹⁵ However, for either method, the storage complexity is not proportional to the size of the bit string. On the other hand, for MSD arithmetic, the storage complexity is linearly proportional to the bit string size. It has been shown that using a direct CAM for each MSD addition output bit, the storage of up to fifty-six holograms is required. In this section, to implement optically the above-mentioned conditional symbolic MSD logic operators, a new CAM method is described. Using two stages of these operators, either MSD addition or subtraction operation can be implemented. With this approach, by reducing the bit-wise memory storage density, practical construction becomes possible.

The first CAM construction step is the minimization of the truth table grouped logic expressions. To express our truth-table results, the sum of product form

Table V. Reduced Logic Minterm Expressions for the Implementation of the MSD Addition and Subtraction Operations. The X Denotes a Do Not Care as Specified in Table VII. The Four Variable Minterm is Grouped as $[X_i Y_i]$.

MSD logic	bit	reduced product term	MSD logic	bit	reduced product term
T	1	$[X_3 X_2] X_1 X_0$ $[X_3 X_2] X_1 X_0$	W	1	$[X_3 X_2] X_1 X_0$ $[X_3 X_2] X_1 X_0$
T	2	$[X_3 X_2] X_1 X_0$ $[X_3 X_2] X_1 X_0$	W	2	$[X_3 X_2] X_1 X_0$ $[X_3 X_2] X_1 X_0$
T	3	$[X_3 X_2] X_1 X_0$ $[X_3 X_2] X_1 X_0$	W	3	$[X_3 X_2] X_1 X_0$ $[X_3 X_2] X_1 X_0$
A	4	$[X_3 X_2] X_1 X_0$ $[X_3 X_2] X_1 X_0$	C	4	$[X_3 X_2] X_1 X_0$ $[X_3 X_2] X_1 X_0$

Table VI. Comparison Among the One-, Two- and Three-Step N -bit CAM MSD Addition Schemes. In Terms of Processing Speed, the One-Step Method is the Fastest, While in Terms of the Product of the Speed and Total Number of References, the Two-Step Method is the Optimum.

methods	No. of CAMs	No. of refs. for each CAM	total No. of refs.	normalized speed
1-step	$N + 1$	≤ 56	$56N - 74$	1
2-step	$3N - 1$	≤ 12	$22N - 4$	2T
3-step	$5N - 3$	≤ 4	$18N - 10$	3T

is chosen. Among the three (1,0,1) possible groups, only two groups are of interest.¹⁵ For the 1 and 1 output groups, using Karnaugh logic minimization maps together with Tables II-IV, in Table V, the reduced logic expressions are shown. Here, for the do not care bits, the notation is either an x for the completely do not care of (1,0,1) or a $x_{0,1}, x_{0,1}$, or $x_{1,1}$ for the partial do not care for the pairs (0,1)(0,1), and (1,1), respectively. In each four variable minterms, the two first (second) column bits denote the variables X_i, Y_i (X_{i-1}, Y_{i-1}), respectively. In the case of Table IV, instead of addressing eighty-one LAM outputs, only six CAM patterns are needed. It can be shown that with the Takagi *et al.* MSD addition truth table,⁹ instead of our eighteen, with a less compact representation, more logic minterms are generated. Also, with this approach, a regular grouping, i.e., three (six) reduced minterm expressions for the $T(W)$ function, for either MSD addition or subtraction, can be obtained. Compared with the MG single-step CAM method and as a result of using the two steps, the addition speed decreases. However, in each step, the storage complexity is drastically reduced. For example, for the addition of two 16-bit numbers, while the MG method requires a storage of 822 minterms, with the new two-step method, only 340 minterms are needed. Thus the new method has the potential to process long bit strings. In Table VI, a comparison between this and some of the other methods are presented. It is noted, while in terms of processing speed, the single-step method is the fastest. However, in terms of the product of the speed and number of reduced minterms, the two-step method is the optimum. Another advantage

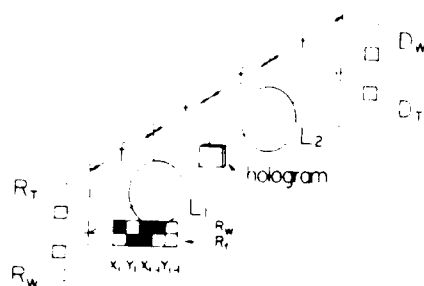


Fig. 2. Holographic CAM recording and logic processing unit. X and Y , the input variables; R_T and R_W , the general reference and the reference for the transfer (weight) operations, respectively; D_W , detector array for logic $T(W)$ operation.

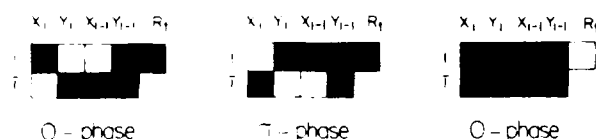


Fig. 3. Holographic CAM-based MSD processing for the pattern 111X. Three recording steps are used.

of the two-step method is that, since the complement of addition (subtraction) can easily be generated, it can be used for error detection.

After this truth-table reduction, an optical memory is constructed. Here an existing optical holographic technique proposed by Mirsalehi and Gaylord¹⁷ is used. In Fig. 2, a CAM-based MSD holographic logic recording and reconstruction system is shown. Each digit is spatially encoded with two vertical pixels. The transparent top (bottom) pixel indicates a logic level 1 (0), while both opaque pixels indicate a logic level zero. For our conditional substitution rule, four $(X_i, Y_i; X_{i-1}, Y_{i-1})$ parallel input bits are used. The other inputs are the general reference R_T and the reference transfer R_T and weight R_W logic bits, respectively. The optical memory consists of a thick Fourier hologram that is able to store a large number of angularly multiplexed spatial Fourier transform patterns. For each required minterm, three recording steps are used. For example, in Fig. 3, the recording of the pattern (111X) for the T operator is indicated. During the first exposure, with the reference bit R_T , the complement pattern (111X) is recorded. In the second step, with a π phase shifted R_T , the input (111X) is recorded. In the last step, with the zero phase R_T the general reference R_T pattern is recorded. If with the first two recordings the bit-wise exposure E is identical, the last step exposure should be mE , where m is the number of second-step nonzero signal digits. These three steps complete the recording process.

For logic operation, the R_T pixel is off, while the signal and R_T pixels are on. For a prerecorded input, the light diffracted by the second and third step recorded patterns cancel (due to their identical magnitude and opposite phase) forming a dark pattern in the detected area. With a different input, the detected

Table VII. Various MSD Do Not Care Types and Their Corresponding Spatial Encoding: m , n , and E Denote the Number of Nonzero, the Number of Do Not Care Bits, and the Single General Reference Bit Exposure, Respectively.

don't care type	patterns to be recorded		R_T -bit exposure
	0-phase	π -phase	
X			mE
X_{0T}			mE
X_{0T}			mE
X_{T1}			$(m+n)E$

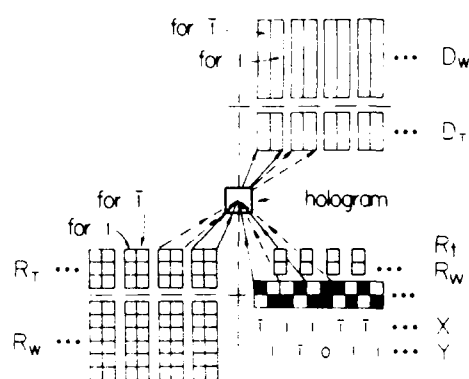


Fig. 4. Schematic setup for implementing the MSD T and W logic operations using a holographic CAM-based system.

residue light is considered as an input mismatch. To record a group of patterns, an angular multiplexing method that employs a group of reference pixels, e.g., the dotted squares in Fig. 2, is needed. It can be shown that the do not care digit can be assigned to the patterns as shown in Table VII, where in the last column n denotes the number of do not care bits. To perform MSD logic operations, both the spatially encoded bit strings and reference bit arrays are used. In Fig. 4, a holographic CAM logic device for implementing the MSD addition (subtraction) operators $T_{+(-)}$ and $W_{+(-)}$ is shown. For each of the three (six) required patterns (see Table V) for $T_{+(-)}$ ($W_{+(-)}$), using this and a spatially encoded reference pixel pattern R_T (R_W), an angularly multiplexed hologram is formed. Also, for the recording and later reading, an array of general reference R_T pixels is used. At the output side, two detector arrays, D_T and D_W , with each divided into two parts for detecting a 1 and 0, are used. The absence of light in one part of a detector indicates a nonzero, while the presence of light on both detectors assigns a zero to the output. These output signals can be used to generate inputs for a next stage logic function.

IV. Summary

To summarize, an alternative method to perform the MSD arithmetic is proposed. First, to rearrange the

number strings to be added or subtracted, MSD addition and subtraction conditional symbolic substitution rules are used. The rearranged numbers are then added, using another bit-wise substitution rule, to produce both the final result and its complement. For an optical implementation, a CAM processing technique is used. First, to obtain a reduced logic (in a sum of minterm form) or equivalently a reduced reference pattern, logic minimization is performed. To record these reduced reference patterns, an angularly multiplexed thick Fourier hologram recording setup is described. The stored holograms optically implement the required logic substitution elements.

Constructive comments by the referees are deeply appreciated. This work was supported in part by a grant from the U.S. Air Force Office of Scientific Research.

References

1. S. L. Hurst, "Multiple Valued Logic—its Status and its Future," *IEEE Trans. Comput.* **C-33**, 1160 (1984).
2. T. T. Dao and D. M. Campbell, "Multiple-Valued Logic: an Implementation," *Opt. Eng.* **25**, 14 (1986).
3. G. Eichmann, Y. Li, and R. R. Alfano, "Optical Binary Coded Ternary Arithmetic and Logic," *Appl. Opt.* **25**, 3113 (1986).
4. N. S. Szabo and R. I. Tanada, *Residue Arithmetic and Its Applications to Computer Technology* (McGraw-Hill, New York, 1967).
5. A. Huang, Y. Tsumda, J. W. Goodman, and S. Ishihara, "Optical Computation Using Residue Arithmetic," *Appl. Opt.* **18**, 149 (1979).
6. D. Psaltis and D. Casasent, "Optical Residue Arithmetic: a Correlation Approach," *Appl. Opt.* **18**, 163 (1979).
7. A. Tai, L. Cindrich, J. R. Fienup, and C. C. Aleksoff, "Optical Residue Arithmetic Computer with Programmable Computation Modules," *Appl. Opt.* **18**, 2812 (1979).
8. A. Avizienis, "Signed Digit Number Representations for Fast Parallel Arithmetic," *IRE Trans. Electron. Comput.* **EC-10**, 389 (1961).
9. N. Takagi, H. Yasuura, and S. Yajima, "High-Speed VLSI Multiplication Algorithm with a Redundant Binary Addition Tree," *IEEE Trans. Comput.* **C-34**, 789 (1985).
10. B. L. Drake, R. P. Bocker, M. E. Lasher, R. H. Patterson, and W. J. Miceli, "Photonic Computing Using the Modified Signed-Digit Number Representation," *Opt. Eng.* **25**, 38 (1986).
11. R. P. Bocker, B. L. Drake, M. E. Lasher, and T. B. Henderson, "Modified Signed-Digit Addition and Subtraction Using Optical Symbolic Substitution," *Appl. Opt.* **25**, 2456 (1986).
12. M. M. Mirsalehi and T. K. Gaylord, "Logical Minimization of Multilevel Coded Function," *Appl. Opt.* **25**, 3078 (1986).
13. C. C. Guest and T. K. Gaylord, "Truth-Table Look-up Optical Processing Utilizing Binary and Residue Arithmetic," *Appl. Opt.* **19**, 1201 (1980).
14. M. M. Mirsalehi, C. C. Guest, and T. K. Gaylord, "Residue Number System Holographic Truth-Table Look-up Processing: Detector Threshold Setting and Probability of Error due to Amplitude and Phase Variations," *Appl. Opt.* **22**, 3583 (1983).
15. M. M. Mirsalehi and T. K. Gaylord, "Truth-Table Look-up Parallel Data Processing Using an Optical Content-Addressable Memory," *Appl. Opt.* **25**, 2277 (1986).

Digital optical isochronous array processing

George Eichmann, Yao Li, Ping Pei Ho, and R. R. Alfano

An optical isochronous array processing method is proposed. An optical isochronous array processor (OIAP) is a local regularly interconnected processing network that employs an array of identical optical processing elements. In an OIAP, incoming isochronous data are parallel processed in a fashion much like a propagating electromagnetic wavefront. For the various applications, the OIAP processing elements and their interconnections can be different. In this paper, various all-optical OIAP elements are considered. Applications ranging from optical binary number multiplication preprocessing to optical matrix algebra as well as to optical residue arithmetic are presented.

I. Introduction

The three major advantages¹ of optics for the modern day signal processing and computation applications are (1) its ability to process large bandwidth signals at ultrahigh speed; (2) its unguided (free-space) wave propagation property; and (3) its lack of interaction between intersecting beams propagating in a linear medium. A combination of these three salient properties can lead to ultrafast parallel optical signal processing and computing. Using linear optics, a good optical analog parallel processing example is an optical spatial Fourier transform. Other optical analog signal processing methods, such as convolution and correlation, are also available. For numerical computation, however, optical analog processing methods cannot offer in general high numerical precision.² To improve the precision, digital calculations need to be used. Thus a parallel ultrafast digital optical computer has long elicited the research interest of optical scientists and engineers.

Among the three parallel computer structures, i.e., vector processor, multiprocessor system, and array processor (AP), the first two are general purpose, while the last belongs to the special purpose computer category. With an AP, data are parallel processed either

synchronously with a global system clock or asynchronously in a data-driven fashion.³ Because these APs offer solutions to a large variety of signal processing problems, there is a considerable interest in the study of their architectures and programming languages. To implement optically an AP, some of its unique features, such as the parallel input-output channels and the spatially local gate interconnections, need to be considered.

In this paper, methods to implement various optical AP (OAP) architectures are proposed. Since these OAPs can process incoming data in an isochronous fashion, that is, whenever the inputs arrive at the same time outputs are generated, the term, optical isochronous array processor (OIAP), is used. Since the IAP is also a locally regularly interconnected network, it is a subset of the systolic AP (SAP). The difference between the two is that in a SAP, the elemental processor is an arithmetic processor (i.e., adder and multiplier), while in an IAP, lower level processors such as a logic gate can be used. With the current optical technology, an OIAP is easier to implement. In the following, for the various OIAP operations, as fundamental processing units regularly interconnected ultrafast nonlinear optical logic elements are proposed. Since some of the processing units can have a femtosecond response, it is possible that these pipelined OIAPs will process data in the picoseconds. The paper is organized as follows: in Sec. II a number of all-optical elemental processing units are briefly discussed. In Sec. III, the use of AND element-based OIAP for optical binary multiplication is described. In Sec. IV, various OIAP-based binary element matrix algebra processors are presented. In Sec. V, fundamental residue mapping units are described, while in Sec. VI, based on these mapping units, an OIAP matrix-matrix residue multiplier is proposed. Finally, Sec. VII summarizes the results of this paper.

The authors are with City University of New York, City College, New York, New York 10031.

Received 17 December 1986.

0003-6935/87/142726-08\$02.00/0.

© 1987 Optical Society of America.

II. Ultrafast OIAP All-Optical Elemental Processing Units

An isochronous wavefront AP is an array of locally interconnected identical processing units. For different data processing applications, the processing unit can be different. In this section, some possible elemental OIAP processing units are described. For the use of an OIAP for arithmetic processing, such as a binary scalar, vector as well as matrix multiplications, an algorithm, the so-called digital multiplication via analog convolution² (DMAC) scheme may be utilized. In this approach, analog optics is commonly used to obtain as the first step a mixed-binary output format convolution result. In principle, for the multiplication of two large numbers, compared to traditional shift/add multipliers, the DMAC processor is faster. However, its actual performance is limited by electronic A-D postprocessing.⁴ With a current version of DMAC binary convolver, the two fundamental processing units are an AND gate and a summer. The summation operation can easily be implemented via a lens. To perform the AND operation, most approaches adopt a hybrid, either an acoustooptic (AO) or an electrooptic (EO), methodology. To increase the processing speed, an all-optical convolution method must be used. In this section, for the OIAP convolution preprocessing step, several ultrafast all-optical elemental processing approaches are proposed.

Currently, there are a number of techniques available to perform an all-optical AND logic operation. However, among these only those AND elements that have a spatially symmetrical input and output channel format are suitable for an AP. This format will not introduce additional time delay, a delay that tends to slow down the computational speed. As possible symmetric input and output channel optical AND elements, four different ultrafast gates are next briefly considered. A simple AND device is a three-input-beam optical nonlinear etalon^{5,6} [see Fig. 1(a)]. The two symmetrical beams *A* and *B* are the logic inputs, while the third (middle) input beam *R* serves as a bias or optical reference. Initially, using the *R* beam, the etalon is tuned to a low power transmission state. Next, the *A* and *B* beam intensities are adjusted so that when all the three beams are on, the total optical power reaches a switching threshold. Above the threshold, the etalon is in a high transmission state switching the signal *R* from the input to the output. The angle θ is adjusted so that no bistable switching phenomena occur to *A* and *B*. This is an important condition since the two off-axis outputs must serve as the inputs to all subsequent OIAP stages. Using a similar geometry, a three (two for logic and one for optical bias) input optical Kerr gate⁷ [see Fig. 1(b)] can also act as a symmetric channel optical logic AND element. There are two differences between the Kerr and etalon AND gate. With Kerr AND gate, the logic input and output have different polarization states, and because this device is not bistable, a clear switching threshold does not exist.

A third candidate is an optical phase-conjugate AND gate⁸ [see Fig. 1(c)]. The beams *A* and *B* act as the two logic inputs, while the beam *R* is a bias (reference)

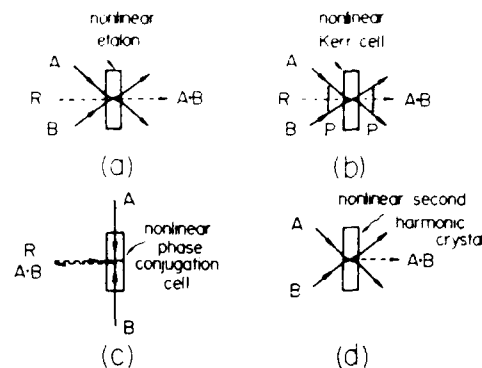


Fig. 1. Four all-optical AND candidates as elemental processing elements for OIAP. *A*, *B*, and *R*, two logic and a bias input beams; *P*, linear polarizer.

source. Due to the nonlinear phase conjugation via degenerate four wave-mixing effect with respect to the reference beam the logic AND output beam travels in a counterpropagating direction. There are other possible all-optical six-port logic AND gate structures, i.e., structures with two logic and one power supply inputs. However, from an OAP implementation point of view, these elements are not ideal, because with an array a large number of optical bias (reference) channels must be established. For this reason, it is better to use a gate that uses only two symmetric inputs without the need of a bias beam. An optical second harmonic generation (SHG) or parametric wave generation via nonlinear optical three-wave mixing^{9,10} AND device does not require a bias beam. A SHG gate has also the potential for a femtosecond response. In Fig. 1(d), a SHG-based AND gate is shown. Two symmetrical identical frequency and polarization logic input beams, denoted *A* and *B* with an angular separation θ , are directed into the SHG crystal. When the angle is adjusted so that the so-called 90° phase-matching condition⁹ is satisfied, in the bisecting input angular direction, a second harmonic (SH) output signal is generated. One advantage of this structure is that, since only a small part of the input energy is converted to a SH output, most of the fundamental input power passes through the gate allowing the outputs to be used as inputs to feed subsequent AND gates. A second advantage is that a number of such AND gates can optically be interconnected on a single SHG crystal. Thus, except for the change in the output frequency, a SHG-based AND gate is a good candidate for an elemental device.

Next, an AND gate-based OIAP operation is discussed. In Fig. 2, a schematic OIAP network where the intersections indicate the AND element placements is shown. The outputs from these elements are marked by dashed lines. To guarantee the isochronous arrival of the two optical signals at each intersection, using either a holographic grating or a composite prism,¹¹ both input wavefronts are tilted at an angle θ (see Fig. 2). The output of each logic AND gate is directed to subsequent logic or memory device for further processing. Using an array of such all-optical

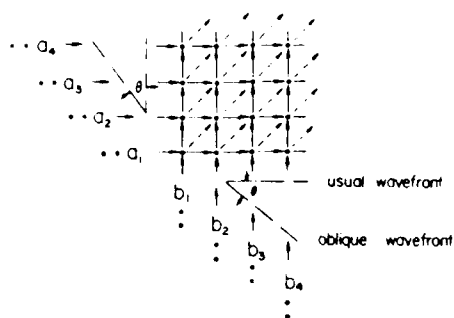


Fig. 2. Schematic of a rectangular OIAP. a_i and b_i are parallel isochronous incoming data. θ is an oblique wavefront angle for synchronizing the input data. An intersection indicates the presence of an identical elemental processing unit generating a dashed line output.

AND gates, an OIAP network can be constructed. For multistage operation, since it is necessary to convert the SH signal back to its fundamental frequency, parametric frequency down (PFD) conversion is needed (see Fig. 3). Using a parametric wave-mixing process with a strong fundamental frequency third harmonic (TH) beam, the SH signal can be converted to the fundamental and amplified. With a KDP crystal, the process has been experimentally demonstrated.¹² In the next sections, the utilization of an AND gate-based OIAP to the implementation of various optical digital arithmetic operations is discussed.

III. Digital Multiplication Using an OIAP

In this section, the computation of both fixed and floating point binary number scalar multiplications, using various OIAP preprocessing networks, is described. In either case, the OIAP is to be used as a convolution preprocessor. Since the convolution results are represented in the mixed-binary format, to complete the operation it must be followed by either an electronic or optical A-D converter^{13,14} and a shift/add (S/A) array. In terms of numerical complexity, the implementation of a highly accurate electronic A-D converter is as difficult as the multiplier itself.⁴ Thus a useful DMAC processor must employ both a highly accurate ultrafast optical convolver and a A-D post-processor. In the following, the ultrafast optical implementation of the first part of a DMAC processor is proposed. For our discussion, irrespective of the practical input crossing angle inside the crystal, the schematics are drawn with all the input channel beams to intersect perpendicularly.

The magnitude of the product of two N -bit binary numbers,

$$A = \sum_{i=0}^{N-1} a_i 2^i \quad B = \sum_{j=0}^{N-1} b_j 2^j$$

can be expressed in two steps as⁴

$$|P| = |AB| = \sum_{i=0}^{2N-1} C_i 2^i \quad (1a)$$

where

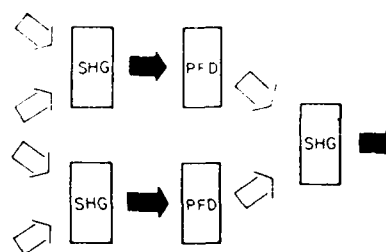


Fig. 3. Staged SHG-based OIAP network with nonlinear parametric frequency-down (PFD) conversion devices as interconnection elements.

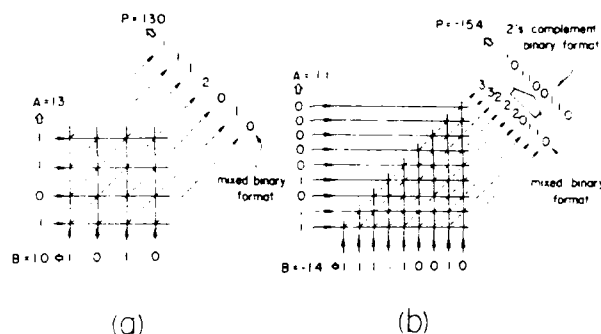


Fig. 4. Two OIAPs for binary (a) unsigned and (b) two's complement signed multiplication preprocessing. Parallel spatially encoded inputs enter from the left and the bottom part of the diagram. Outputs generated at each AND gate beam intersection are used to form the mixed-binary multiplication result.

$$C = \sum_{k=0}^{N-1} a_k b_{N-1-k} \quad (0 \leq k \leq N-1) \quad (1b)$$

is the i th digit weight of resulting product in a mixed-binary format.² Equation (1b) can also be interpreted as a i th digit result from the two number sequence convolution. To obtain this multiplication, using the DMAC algorithm the binary convolution indicated in Eq. (1b) needs to be first performed. This convolution in a real-time scheme uses sequential electronic signals to drive two cascaded acoustooptic (AO) Bragg cells.^{15,16} Using integrating lenses, the deflected wavefronts are collected at their corresponding detectors. Because of the serial input, the convolution of two N -bit words requires $2N - 1$ computation cycles. The cycle time is mainly determined by the speed of the acoustic wave. To increase the convolution speed, next, an OIAP parallel digital convolution scheme is proposed.

For implementing a two 4-bit binary number parallel multiplication, consider a 4×4 optical AND gate array [see Fig. 4(a)]. At each beam intersection, an optical logic AND element is placed. The input bit spacings D_A and D_B are identical. The seven parallel channel AND gate outputs, indicated by dashed lines, represent the seven output digits. For example, to multiply two unsigned decimal numbers $P = A \times B = 13 \times 10$, first, each decimal number is converted into its corresponding binary format. In this case, they are $A = 1101$ and $B = 1010$. These bits are then spatially

optically encoded as light pulses to be launched into the convolution network. The number of cumulative AND output pulses in seven output channels yields the mixed-binary number convolution result 1112010. This mixed-binary number corresponds to the decimal number 130.

To include sign information, a two's complement binary (TCB) representation^{17,18} can be used. In a TCB representation, an additional 0 (1) sign bit in front of the most significant bit (MSB) represents plus (minus) sign information. For a positive number, it simply places a zero in front of the MSB in its unsigned binary form. To obtain a negative number, its positive counterpart is first complemented, and then to it a one is added. For example, the signed decimal numbers $A = 11$ and $B = -14$ have as their TCB representations the numbers $A = 01011$ and $B = 10010$. It has been shown¹⁸ that to multiply two 5-bit TCB numbers, both nine input and output channels are required. For this reason, four zeros (one) are inserted between the sign and MSB of $A(B)$. As an example, to multiply these TCB numbers, in Fig. 4(b), a SHG-based TCB multiplier is shown. The network takes inputs A and B and generates in parallel the mixed binary output $P = 332220110$. To convert this number back to its TCB format, the least significant (LSB) bit is first divided by two, and the quotient is added to the next bit and so on. The result is $P = 101100110$ representing the decimal number -154 . Since this TCB multiplication network uses the lower triangular half of the previous unsigned binary multiplication network, two SHG-based implementations can be used. Either only part of the unsigned number multiplication network output (corresponding to lower triangular array) channels is used or the upper triangular half crystal is replaced by an index-matching liquid.

The previously described multiplication methods use, both for the input and output, a fixed point binary representation. Recently, based on the DMAC algorithm, an optical floating point binary multiplication scheme, also known as a fixed point number multiplication,¹⁹ has been suggested. Based on the SHG AND gate array, next, the operation of a fixed point unsigned binary multiplier is described. Assume, as an example, the numbers to be multiplied are $A = 7/2$ and $B = 5/32$. The corresponding floating point binary representations are $A = (0.111)2^{+2}$ and $B = (0.101)2^{-2}$. For the use of 3-bit mantissas and the two exponents, $+2$ and -2 , with the fixed point binary representation, each number is represented by seven binary digits, i.e., $A = 11.10000$ and $B = 00.00101$. These digits represent the parallel inputs to the previously described SHG-based unsigned binary convolver network. At the convolver output, the train of pulses $P = 0000112110000$ is generated. After setting the decimal point¹⁹ that can be performed electronically the final result is $P = 0.011211$, representing the correct fraction $35/64$. Since, with the fixed point multiplication technique, the OIAP is used for handling mantissas, the method can directly be extended to perform a signed floating-point TCB multiplication.

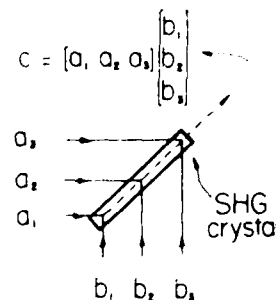


Fig. 5. OIAP 3-D vector inner product processor. Three AND-gate outputs are aligned in a single output channel.

IV. Algebraic Processing Using SHG-Based OIAP Networks

The normal method to multiply two binary numbers requires three operations: a logic AND, a shift, and an arithmetic sum operations. With the DMAC scheme, the multiple shift and sum operations are bypassed by using a mixed-binary representation. The mixed-binary representation allows the successive addition of several numbers before a final A-D conversion. More complicated algebraic operations such as matrix algebra can also be decomposed into several multiplications and additions that can then be performed in parallel. For performing digital optical matrix algebra, several DMAC-based architectures have been proposed.¹⁵⁻²⁰ In this section, using an OIAP, various optical binary algebraic operations, such as vector-vector, matrix-vector, as well as matrix-matrix multiplications, are described.

Given two N -dimensional (N -D) column vectors

$$A = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} \quad B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} \quad (2)$$

the inner or scalar (dot) product of the two vectors is defined as

$$C = A \cdot B = [c_1, c_2, \dots, c_N] \quad c_i = a_1b_1 + a_2b_2 + \dots + a_Nb_N \quad (3)$$

Assuming all $a_i(b_j)$, where $i, j \in (1, 2, \dots, N)$ are either a zero or one, Eq. (3) can be implemented with N number of AND gates and a summer. As an example, in Fig. 5 the inner product of two 3-D vectors is considered. Two 3-bit parallel inputs are directed to a SHG crystal. The three intersection AND gates are aligned so that their outputs can be routed into a single channel. Using a time-integrating output detector, the detected inner product result is in a mixed binary form. This result can then be converted using an A-D device to its binary form.

The vector outer product of two N -D binary vectors, A and B , is defined as the matrix

$$[C] = AB^T = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} [b_1, b_2, \dots, b_N] = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1N} \\ c_{21} & c_{22} & \dots & c_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ c_{N1} & c_{N2} & \dots & c_{NN} \end{bmatrix} \quad (4)$$

where $c_{ij} = a_i b_j$. To perform this vector outer product multiplication, N^2 AND gates are needed. In Fig. 6, a schematic network illustrating the multiplication of two 3×3 vectors is shown. To expand the input light dots into either horizontal or vertical light bars to cross and overlap at the SHG plate, two additional cylindrical lenses are employed. The SH signals emanating from the nine intersections are considered as the outer product outputs.

The matrix-vector product of a $N \times N$ binary matrix $[A]$ and a N -D column vector B is defined as

$$D = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} = \begin{bmatrix} a_{11}b_1 + a_{12}b_2 + \dots + a_{1N}b_N \\ a_{21}b_1 + a_{22}b_2 + \dots + a_{2N}b_N \\ \vdots \\ a_{N1}b_1 + a_{N2}b_2 + \dots + a_{NN}b_N \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} \quad (5)$$

Note that a matrix-vector product can be decomposed into several parallel vector inner product operations. In Fig. 7, by combining with an additional input cylindrical lens, three Fig. 5 type networks, a SHG-based 3-D vector optical matrix-vector multiplier is shown.

Compared to the previously described algebraic operations, an optical matrix-matrix multiplication is more complicated. As an example, consider the multiplication of two 2×2 binary matrices $[A]$ and $[B]$

$$[E] = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} \quad (6)$$

There are two methods to evaluate this matrix-matrix product. Using a vector outer product decomposition, the matrices are first decomposed into vectors and then are sequentially entered into a physical vector outer product multiplier. With a properly decomposed synchronized temporal sequence, the previously described outer product processor (see Fig. 6) can be used for the matrix-matrix product generation. Using a vector inner product decomposition in combination with an AND gate-based OIAP provides a faster optical matrix-matrix multiplier. In Fig. 8, a vector

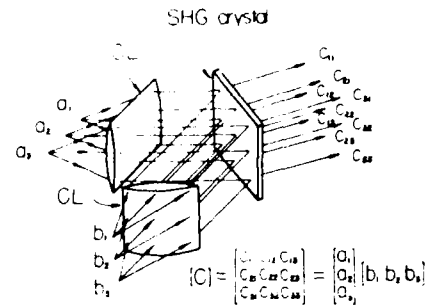


Fig. 6. OIAP 3-D vector outer product processor. In addition to nine SHG AND gates, two input prisms are used for interconnections.

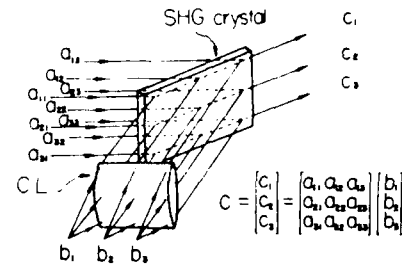


Fig. 7. OIAP matrix-vector multiplier. Three 3-D parallel vector inner product processors together with an input prism are used.

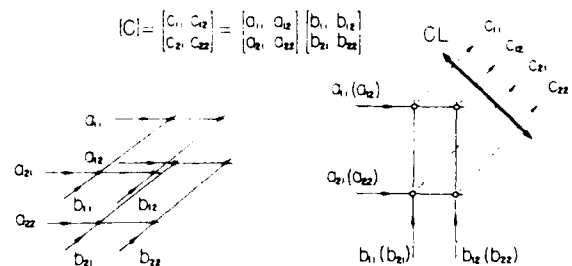


Fig. 8. OIAP matrix-matrix multiplier. Parallel input spacings for $[A]$ and $[B]$ are different so that four output channels (after a lens space integration) are obtained.

inner product-based OIAP matrix-matrix multiplier is shown. Unlike the previous examples, both a 3-D network and unequal input channel spacings are used. In this network instead of entering the data sequentially as in the case of a vector outer product multiplier,²⁰ a fully parallel input format is used. Because of the parallel format, it possesses an inherently faster multiplication speed. To collect the channelized multiplication results at the output an additional cylindrical lens is used.

V. Fundamental Residue Mapping Units using OIAP

The major attraction of the residue number system is its carry-free arithmetic operation capability.^{21,22} In this system, to perform numerical calculation, a set of relative prime integer bases is used. Each base forms a group of independent parallel processing units.

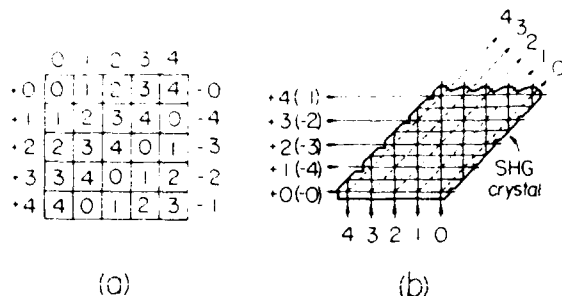


Fig. 9. (a) Mod5 residue addition (subtraction) truth table. (b) SHG-based mod5 residue optical adder implementation. At the interfaces, total internal reflections are used.

To obtain the result in a specific modulo, in each independent residue channel, various mapping functions are performed. By suitably combining the partial results generated from each processing unit the final result is formed. Thus a specific modulo processing unit is the key element to implement a residue-based numerical processor. Using electrooptic waveguide switches, different modulo residue mapping units have been proposed.²³ Next, several SHG-based all-optical residue mapping units are described and then used to implement a residue-based matrix multiplication.

In a mod N residue number system the addition is a circulant operation with the number of integral shifts determined by the value of addend. Similarly, subtraction is also a circulant operation where the subtrahend is added but in an opposite direction. To illustrate this principle, consider a mod5 addition/subtraction truth table [see Fig. 9(a)]. To implement this table, a 2-D optical mapping network must be constructed. In Fig. 9(b), using a SHG crystal, such a OIAP network is shown. For both the summand (minuend) and addend (subtrahend), pulse-position coded inputs are used. The mapping operation is controlled by the summand (minuend) signals. Thus generated signals share five output channels. The crystal is cut and oriented so that desired total internal reflections can be achieved. Here at any given time only one of the five addend (subtrahend) channels contains a signal. In other words, for a pair of input pulses, a SH output can only be generated at a single intersection. Again, using an oblique input isochronous wavefront, the inputs are autosynchronized, and thus no additional delay elements or clocks are needed.

In addition to residue addition/subtraction, using an OIAP, mod N residue multiplication can also be performed. A mod5 multiplication truth table, for example, is shown in Fig. 10(a). Since the multiplication by a zero results in a zero, only operations that map the other numbers, i.e., 1, 2, 3, and 4, are necessary. For this reason, to implement a mod5 multiplication truth table, use of a mod4 adder has been suggested.²³ In general, multiplication in mod p , where p is a prime, can be decomposed into addition in mod $(p-1)$ with suitable prepermutation and postpermutation networks. In Fig. 10(b), using a SHG-based OIAP net-

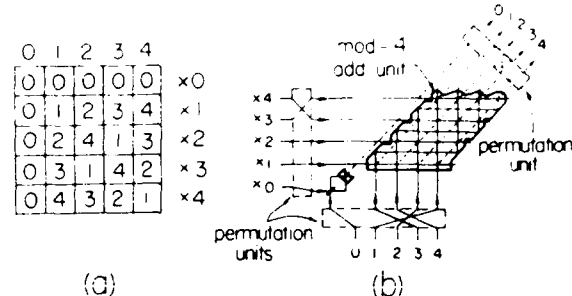


Fig. 10. (a) Mod5 residue multiplication truth table. (b) SHG-based mod5 optical multiplication implementation. Two SHG crystals, one for a mod4 addition and another to handle the zero, are shown. Also, a prepermutation and a postpermutation devices are also used.

work, an SHG-based residue mod5 multiplication unit is shown. In addition to three input (output) permutation elements, two separate SHG units, one for implementing mod4 add and another for dealing with zeros, are employed. Although a multiplier uses more elements than a corresponding add unit, compared to other methods²³ this is a faster and more compact unit.

VI. Residue-Based OIAP for Matrix Multiplication

In many real-time scientific and engineering problems, it is necessary to solve a large number of algebraic and differential equations. For the solution of these equations, a large amount of matrix manipulations are needed. With a digital optical computer, it is essential to be able to perform fast matrix multiplication. In previous sections, using the DMAC algorithm, an AND element-based binary matrix multiplication preprocessor was described. In this section, using residue arithmetic, an alternative integer matrix-matrix multiplication approach is proposed. Using a residue number system, integer matrix multiplication can be decomposed into a set of parallel relative prime modulo-based residue matrix multiplications. Thus solving a set of linear equations using residue matrix algebra can increase computational speed. Details on the solution of integer-valued linear equations using residue matrix algebra are available.^{24,25}

The multiplication of two identical prime modulo-based matrices is similar to the decimal case. In the residue case, both multiplication and addition are evaluated in the specific modulo residue system. A conventional matrix-matrix multiplier that performs the multiplication of two $N \times N$ matrices ($[C] = [A][B]$) contains a 2-D square array of N^2 identical processing elements, each performing an arithmetic/logic operation that adds to its past contents the multiplication results of two present inputs. For the multiplication of two mod p $N \times N$ matrices, each of the N^2 cells, for example, C_{ij} , where $i, j \in (1, 2, \dots, N)$, executes recursively for $k = 1, 2, \dots, N$,

$$[C_{ij}^{(k)}] = [C_{ij}^{(k-1)} + a_{ik}b_{kj}] \text{ mod } p, \quad (7)$$

where $a_{ik}(b_{kj})$ are the ik th (kj th) element of the matrix

$[A]([B])$. As an example, consider the multiplication of two $4 \times 4 \text{ mod } 5$ matrices $[A]$ and $[B]$, where

$$[A] = \begin{bmatrix} 1 & 2 & 0 & 3 \\ 4 & 1 & 0 & 2 \\ 3 & 0 & 2 & 4 \\ 2 & 2 & 1 & 0 \end{bmatrix} \quad [B] = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 2 & 4 & 0 & 2 \\ 3 & 0 & 1 & 3 \\ 2 & 3 & 4 & 1 \end{bmatrix} \text{ mod } 5. \quad (8)$$

Each of the sixteen processing units performs identical arithmetic operations, i.e., $\text{mod } 5$ -based multiplications and additions. It can be shown that the corresponding matrix multiplication result is

$$[C] = [A][B] = \begin{bmatrix} 1 & 3 & 3 & 2 \\ 0 & 4 & 2 & 4 \\ 2 & 0 & 1 & 0 \\ 4 & 0 & 3 & 2 \end{bmatrix} \text{ mod } 5. \quad (9)$$

To implement this 4×4 residue matrix multiplication using an OIAP the array shown in Fig. 11(a) may be used. Input arrays A and B enter from the left and the top part of the processor. To guarantee the isochronous data arrival, a number of zero's are used. In Fig. 11(b), an individual processing element, a $\text{mod } 5$ multiplier/accumulator, is depicted. White and black arrows represent the input and SH beams. The residue multiplier performs on the two present inputs a $\text{mod } 5$ multiplication. Its result is first converted from a SH to a fundamental frequency and then added using a $\text{mod } 5$ adder to the adder's previous content. The optical delay line is adjusted so that the residue adder previous content arrives isochronously with the present multiplier output. After four recursions, at each element C_{ij} the desired output is generated. Using such residue OIAPs as building blocks, an all-optical matrix residue processor may be constructed.

VII. Summary and Conclusion

In this paper, various all-optical array processing methods have been introduced. For OIAP implementations, several nonlinear optical devices, such as a nonlinear etalon, Kerr gate, optical phase conjugator, and SHG device, were described. Among various proposed all-optical AND elements, the SHG AND gate is a preferred candidate. A SHG AND gate can have a femtosecond response. Except for a frequency change, a SHG-based OIAP can be monolithically implemented and thus possesses the potential for optical circuit integration. For multiple-stage SHG-based OIAP operation, using PFD techniques, the doubled frequency can be converted back to its fundamental frequency. The spatially encoded 2-D parallel data are processed in a locally interconnected lattice-type OIAP network. For digital optical arithmetic computing, a binary AND gate-based OIAP is discussed. In principle, for different applications, other elemental logic and arithmetic operators can also be used. In Table I, the use of different AND element OIAP arrays, where the dot and circle represent 2- and 3-D arrays, respectively, is summarized. Parallel data are assumed to enter from both the left and bottom of the

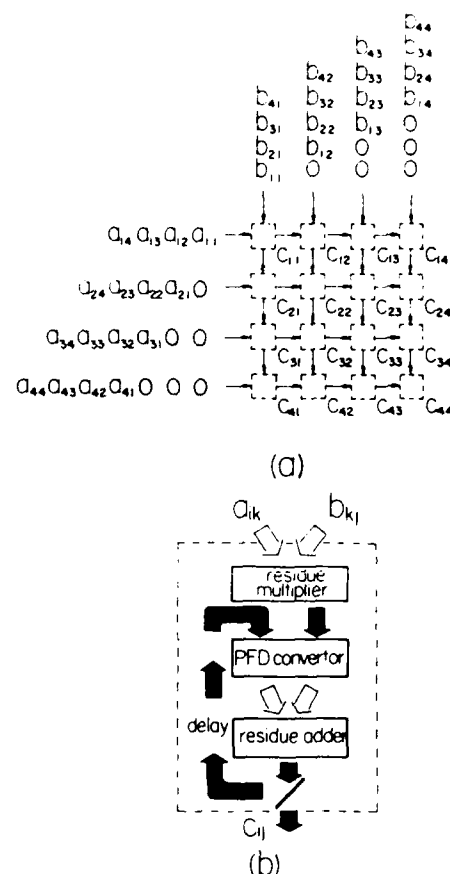


Fig. 11. (a) A 4×4 residue matrix-matrix multiplier with sixteen identical arithmetic processing cells C_{ij} . (b) A SHG-based C_{ij} cell that employs a $\text{mod } 4$ multiplier and a $\text{mod } 4$ adder. Black and white arrows represent the SH and fundamental frequency beams. In addition to the adder and multiplier, a PFD conversion device is needed.

Table I. Summary of the Various Proposed Arithmetic OIAPs

unsigned binary multiplication	2's complement multiplication	residue logic	vector-vector inner-product
vector-vector outer-product	matrix-matrix product	matrix-vector product	

Note: Each dot indicates an elemental processor. The black and white dots are the 2- and 3-D arrays for the different computations. Parallel data enter the processor from both the left and bottom of the diagram.

network. Using these structures with either a DMAC or a residue algorithm, various optical multipliers were described.

This work was supported in part by a grant from the U.S. Air Force Office of Scientific Research 84-0144.

References

1. A. Huang, "Architectural Considerations Involved in the Design of an Optical Digital Computer," *Proc. IEEE* **72**, 780 (1984).
2. D. Psaltis, D. Casasent, D. Neft, and M. Carlotto, "Accurate Numerical Computation by Optical Convolution," *Proc. Soc. Photo-Opt. Instrum. Eng.* **232**, 151 (1980).
3. S. Y. Kung, K. S. Arun, R. J. Gal-Ezer, and D. V. B. Rao, "Wavefront Array Processor: Language, Architecture, and Applications," *IEEE Trans. Comput.* **C-31**, 1054 (1982).
4. D. Psaltis and R. Athale, "High Accuracy Computation with Linear Analog Optical Systems: a Critical Study," *Appl. Opt.* **25**, 3071 (1986).
5. P. W. Smith and W. J. Tomlinson, "Bistable Optical Devices Promise Subpicosecond Switching," *IEEE Spectrum* (26 June 1981).
6. J. L. Jewell, Y. H. Lee, M. Warren, H. M. Gibbs, and N. Peyghambarian, "3-pJ, 82-MHz Optical Logic Gates on a Room-Temperature GaAs-AlGaAs Multiple-Quantum-Well Etalon," *Appl. Phys. Lett.* **46**, 918 (1985).
7. E. P. Ippen and C. V. Shank, "Picosecond Response of a High-Repetition-Rate CS₂ Optical Kerr Gate," *Appl. Phys. Lett.* **26**, 92 (1975).
8. T. R. O'Meara, D. M. Pepper, and I. O. White, in *Optical Phase Conjugation*, R. A. Fisher, Ed. (Academic, New York, 1983).
9. C. Kolmeder, W. Zinth, and W. Kaiser, "Second Harmonic Beam Analysis, a Sensitive Technique to Determine the Duration of Single Ultrashort Pulses," *Opt. Commun.* **30**, 453 (1979).
10. J. Sauteret, T. Duvillier, and A. Adolf, "Harmonic Generation with Non-collinear Laser Beams; Application to Pulse Stacking," *Opt. Commun.* **44**, 135 (1982).
11. Zs. Bor and B. Racz, "Group Velocity Dispersion in Prisms and its Application to Pulse Compression and Traveling-Wave Excitation," *Opt. Commun.* **54**, 165 (1985).
12. R. A. Baumgartner and R. L. Byer, "Optical Parametric Amplification," *IEEE J. Quantum. Electron.* **QE-15**, 432 (1979).
13. H. F. Taylor, M. J. Taylor, and P. W. Bauer, "Electro-optical Analog-to-Digital Conversion Using Channel Waveguide Modulators," *Appl. Phys. Lett.* **32**, 559 (1978).
14. R. A. Becker, C. E. Woodward, F. J. Leonberger, and R. C. Williamson, "Wide-band Electrooptic Guided-Wave Analog-to-Digital Converters," *Proc. IEEE* **72**, 802 (1984).
15. H. J. Caulfield, W. T. Rhodes, M. J. Foster, and S. Horvitz, "Optical Implementation of Systolic Array Processing," *Opt. Commun.* **40**, 86 (1981).
16. P. S. Guilfoyle, "Systolic Acousto-Optic Binary Convolver," *Opt. Eng.* **23**, 20 (1984).
17. R. P. Bocker, S. R. Clayton, and K. Bromley, "Electrooptical Matrix Multiplication Using the Two's Complement Arithmetic for Improved Accuracy," *Appl. Opt.* **22**, 2019 (1983).
18. A. P. Goutzoulis, "Systolic Time-Integrating Acoustooptic Binary Processor," *Appl. Opt.* **23**, 4095 (1984).
19. R. P. Bocker, M. E. Lasher, W. J. Miceli, R. H. Patterson, and B. L. Drake, "Optical Fixed-Point Arithmetic," *Proc. Soc. Photo-Opt. Instrum. Eng.* **564**, 150 (1985).
20. R. A. Athale and J. N. Lee, "Optical Processing Using Outer-Product Concept," *Proc. IEEE* **72**, 931 (1984).
21. S. L. Hurst, "Multiple Valued Logic—Its Status and its Future," *IEEE Trans. Comput.* **C-33**, 1160 (1984).
22. A. Huang, Y. Tsunoda, J. W. Goodman, and S. Ishihara, "Optical Computation Using Residue Arithmetic," *Appl. Opt.* **18**, 149 (1979).
23. A. Tai, I. Cindrich, J. R. Fineup, and C. C. Aleksoff, "Optical Residue Arithmetic Computer with Programmable Computation Modules," *Appl. Opt.* **18**, 2812 (1979).
24. M. Newman, "Solving Equations Exactly," *J. Res. Natl. Bur. Stand. Sect. B* **71**, 171 (1967).
25. P. R. Beaudet, A. P. Goutzoulis, E. C. Malarkey, and J. C. Bradley, "Residue Arithmetic Techniques for Optical Processing of Adaptive Phased Array Radars," *Appl. Opt.* **25**, 3097 (1986).

Optical artificial intelligence and symbolic computing: an introduction

George Eichmann, H. John Caulfield, and Ivan Kadar

After a short introduction, a brief overview of the research papers contained in this special issue is presented.

The notion of what is considered intelligence and *inter alia* artificial intelligence (AI) is a metaphysical issue. As a metaphysical hypothesis it borders on biological, philosophical, sociological, cognitive, and psychological questions about the nature of our universe. As scientists, however, we can define the field operationally as a discipline devoted to the machine solution of a particular set of problems. Some of the commonly mentioned problems that fall in the domain of AI are game playing, mathematical theorem proving, the problems of both speech and vision perception, natural language understanding, and expert problem-solving systems. The underlying theme for the solution to some of these problems involves such concepts as knowledge, reasoning, learning, adapting, organizing, planning, predicting, and understanding—precepts that humans should follow without fail.

To represent these concepts in a machine understandable form, suitable models of these processes must be found. In as much as the processes themselves are still under study, the models that represent these concepts are still in a development stage. Presently, there are three major thrusts in AI model building. In the first school of thought, the AI model is constructed from a set of rather primitive but fundamental operators with built-in rules, the collection of which constitutes a language, that will allow these systems to acquire knowledge through some form of

learning mechanism. These rules allow the language system to reason, learn, adapt, and in general evolve into a more complex system. Examples of some of these language systems are SETL (a set theoretic language), aspects of predicate calculus, and RAFAL (a generalization of LISP). The difficulty with these systems is that to acquire new concepts they require an inordinate amount of machine computation. Another school of thought revolves around the idea of an expert system. An expert problem-solving system does not start from basics but rather codifies the results of an evolutionary process as rules derived from the knowledge of experts. By consulting the knowledge of the experts, the machine is able to mimic a form of intelligence. Finally, since biological systems are living examples of intelligent systems, the mathematical models of biological systems when appropriately coded can, therefore, be considered for AI modeling.

Traditionally, optics deals with the generation, propagation, and detection of light waves. New developments in laser sources and better optical detection devices, however, opened new nontraditional areas of applications of optical devices and techniques. Large aperture optics, with its promise of large-scale parallelism, and the development of picosecond and femtosecond optical switching devices have thrust optics into the area of both optical arithmetic and logic computing. Most optical encoding methodology transforms the arithmetic or logic variables into spatial patterns. The 2-D spatial pattern manipulation is where optics has a distinct advantage over the traditional 1-D electronic processors. A large part of AI computing deals with symbol manipulation. For a 1-D electronic computer implementation of the AI symbol manipulation, the symbols must be converted into strings of digits. However, humans do symbol manipulation using visual images. Thus it is expected that an optical machine vision system can play a large role in modeling AI systems.

The papers in this issue are organized in the follow-

George Eichmann is with City University of New York, City College, Department of Electrical Engineering, New York, New York 10031; H. J. Caulfield is with University of Alabama in Huntsville, Center for Applied Optics, Huntsville, Alabama 35899; and I. Kadar is with Grumman Corporation, Bethpage, New York 11714-3582.

Received 6 March 1987.

0003-6935/87/101827-02\$02.00/0.

© 1987 Optical Society of America.

ing sequence. The first set of papers deals with fundamental issues of modeling AI systems. In the next set of papers, neurological models and applications of neural models to various optical, i.e., visual, application tasks are analyzed. Next, some optical implementation examples of particular expert systems are provided. Finally, possible electrooptical implementations of biologically motivated artificial neural networks are described.

In the first paper in this special issue, "Massive Parallelism in Artificial Intelligence," by L. Shastri, the need for massive parallelism for AI computation is documented. In particular, details of a connectionist model of an AI computing network are described. This network model in turn is applied to problems in vision and natural language processing.

In "Relational Matching," by L. G. Shapiro and R. M. Haralick, another vision-oriented AI problem is discussed. Here the problem is given a machine described image representing a picture of a scene. The question is how do we determine the objects and their spatial relationship. The solution, called relational matching, is a high-level or scene interpretation process. Several sequential and some possible parallel algorithms are described.

In "Optical Implementations of Mathematical Resolution," by R. A. Schmidt and W. T. Cathey, an optical implementation of an important problem in AI is described. Mathematical resolution is a technique where facts are expressed in a clause form. Using quantified propositional calculus, the operations of resolvent formation, unification, and search are performed. A parallel formulation of this method and some optical implementations are described.

The next paper, "Parallel Optical Syntactic Pattern Recognizers," by G. Eichmann and S. Basu, discusses a pattern recognizer where the pattern's structural information is taken into account. In this technique, the pattern is represented by the syntax of a grammar. A parallel syntactic pattern recognition algorithm and some optical implementation approaches are presented.

In "Masking Fields: A Massively Parallel Neural Architecture for Learning, Recognizing and Predicting Multiple Grouping of Patterned Data," by M. A. Cohen and S. Grossberg, a new neural model of associative pattern classification is detailed. The neural model, called the masking field, is applicable to visual object recognition, speech recognition, and cognitive information processing. The masking field recognizes, predicts, scales, adapts, and learns within context information on a multiple grouping of input patterns.

In "Nearest Matched Filter Classification of Spatio-temporal Patterns," by R. Hecht-Nielsen, a particular subset of the masking field approach is discussed. Here a bank of neural filters, each of which contains a large number of pattern classifiers, is employed. These nearest-neighbor matched filter classifiers have near-Bayesian performance and will have applications in speech, sonar, radar, and advanced communication

environments.

In "A Holographic Associative Memory with Non-linearities in the Correlation Domain," by Y. Owechko, G. J. Dunning, E. Marom, and B. H. Soffer, optical implementation of a continuous associative neural network is described. Here a shift-invariant all-optical holographic associative memory is implemented using phase conjugate mirrors and Fourier transform holograms. By using nonlinearities in the correlation domain, crosstalk and correlation noise are reduced.

In "Superresolving Signal and Image Restoration using a Linear Associative Memory," by G. Eichmann and M. Stojancic, the long-standing problem of the superresolving reconstruction (restoration) of an object of known finite spatial extent with the low space-spatial bandwidth product is considered. By proper training of a constrained linear associative memory filter, superresolving signal and image restoration is shown to be achieved.

In "Associative Network Applications to Low-Level Machine Vision," by J. M. Oyster, F. Vicuna, and W. Broadwell, applications of a neural network model to low-level machine vision tasks are detailed.

In "Real-Time Optical Expert Systems," by A. D. McAulay, the advantages of optics for the construction of a real-time expert system are enumerated. In particular, an optical expert control expert may use an optical spatial light modulator as a crossbar switch for interconnecting logic units and may make deductions from incoming data at high speed. As an example, a medical expert system is described.

Using a different approach, in "Model-Based Knowledge-Based Optical Processors" by D. Casasent and S. A. Liebowitz, a model-based optical expert system is described. Here the ability to on-line generate a 2-D image projection or range image for any object/viewer orientation is addressed. The generation of the knowledge base and the applications of this knowledge base in associative processors and symbolic correlators are described.

The paper, "Optical Expert Systems," by H. H. Szu and H. John Caulfield, deals with constructing a page-oriented holographic data base together with spatial light modulators to form an optical expert system.

Finally, in "Photoneural Devices and Architectures," by R. V. Jones, optoelectronic arrays of monolithically integrated autonomous photoneural processors are described. Neural architectures consisting of a layered hierarchy of 2-D integrated arrays of optoelectronic cells form the various perceptual neural processing networks.

The field of artificial intelligence and symbolic computing, as opposed to optics, is new. Many changes are taking place in both of these fields. It is the hope of the special issue editors that the papers presented capture the flavor of this new synergism of the two great dynamic disciplines. It is clear that a lot remains to be learned and developed. It is our hope that this issue will point to new directions for the cross-disciplinary field of optical artificial intelligence and symbolic computing.

Parallel optical syntactic pattern recognizers

George Eichmann and Sanghamitra Basu

There are many pattern recognition problems where the pattern's structural information is important. In these problems, a syntactic method of pattern recognition is of value. In this paper, both parallel syntactic pattern recognition algorithm and optical architecture implementation approaches are described. In particular, the applications of syntactic pattern recognition algorithm to shape classification are illustrated. A number of parallel optical syntactic pattern coding methods, a structural matched filter and associative memory filter, and an optical symbolic substitution syntactic parser are discussed.

1. Introduction

There are many ways to describe a pattern. In a typical approach, each input pattern is converted into a binary representation using a photosensitive matrix device. The sensed data, derived from a measurement grid, are sometimes arranged in the form of a pattern vector $\mathbf{X} = (x_1, x_2, \dots, x_n)^T$, where T is the transpose. In general, the component x_n is either a binary number or a sampled value of a continuous function. It is from this pattern vector that one starts the analysis and gathers information about the pattern.

It has been observed that in some cases a pattern is a collection of many repetitive subpatterns appearing in a certain order. The simplest subpattern is called a pattern primitive. The special order of primitive sequence is completely defined by the syntax of a grammar. Some patterns can be represented by a string of primitives. Every string is a sequence of primitives. The collection of strings, called a language, can be generated by a grammar. String representations are adequate for describing patterns whose structure is based on a relatively simple connection of primitives. For many applications, more powerful approaches are realized through the use of higher-dimensional representations^{1,2} such as trees, webs, and graphs. However, this discussion is restricted to string representation.

The many different mathematical techniques used to solve the pattern recognition problem may be grouped into two general categories: the decision-

theoretic and the syntactic approaches. The decision-theoretic approach³ uses statistical decision and estimation techniques to describe patterns. However, this technique makes very little use of the structure of the patterns. In some pattern recognition problems, the structural information that describes each pattern is important, and successful classification can only be achieved by using this information. Examples for problems where the syntactic method is a suitable approach are fingerprint identification, character recognition, chromosome identification, image segmentation, and object recognition.^{2,4-7}

In a syntactic pattern recognition system (SPRS)² any input pattern first enters the preprocessing stage where it is encoded, smoothed, and enhanced, followed by a pattern representation stage where it is converted into a string of primitives. In the third and final stage, the input string is syntax analyzed by a parser. It is at this step where the input pattern is either rejected or accepted. In general, because of the time the parser takes to analyze a string, the SPRS is a very slow process. We describe parallel context-free language (CFL) recognition algorithms which are suitable for parallel implementation.

The specific tasks involved in the syntactic analyzer are CFL recognition and error-correcting CFL recognition. The recognizer designed for a particular grammar recognizes only those classes of pattern that correspond to this grammar. For example, corresponding to m classes of patterns w_1, w_2, \dots, w_m , m grammars G_1, G_2, \dots, G_m can be constructed so that the strings generated by the grammar G_i represent the pattern in class w_i . For an unknown pattern described by the string x , the problem of recognition is essentially reduced to the answer to the question

$$x \in L(G_i) \text{ for } i = 1, \dots, m?$$

The process resulting in an answer to such a question with respect to a given grammar is called syntax analy-

The authors are with City University of New York, City College, New York, New York 10031.

Received 14 July 1986.

0003-6935/87/101859-07\$02.00/0.

© 1987 Optical Society of America.

sis or parsing.² In addition to answering such a question, the process can also provide the generation of the derivation tree of x , the structural information of the corresponding pattern.

We may describe an unknown pattern by its relational tree.² The relational tree for each reference pattern can be coded by a suitable method into a string and then stored in a memory. The relational tree for the unknown pattern can also be coded into a string using the same method. This string can then be matched against the reference strings.

A second approach, known as the Earley algorithm, is the most commonly used SPRS recognizer.⁸ Although the Earley algorithm is suitable for a general CFL, it is slow with time complexity $O(n^3)$, where n is the length of the string. Graham *et al.*⁹ derived from the Earley algorithm a new on-line CFL recognition algorithm that allows $O(n^2/\log n)$ operations on bit vectors of length n or $O(n^3/\log n)$ operations on a RAM. About the same time Weicker¹⁰ came up with a similar result. The Earley algorithm can be applied to error-correcting CFL recognition.² Because of the repeated error checking and correcting, the error-correcting parser also has been very slow.

Optical pattern recognition (OPR) is generally dated from the publication of the VanderLugt complex matched filter.¹¹ These filters can be implemented with either a coherent or an incoherent optical system. These filters work directly on the physical image. It was found, however, that these filters are rather sensitive to topological as well as other changes in the physical image. In particular, rotation,¹² scaling,¹³ interclass and intraclass pattern variations, etc. may cause misclassification. To extend the operating range of the matched filter, several optical decision-theoretical methods were described.¹⁴⁻¹⁶ Both an OPR overview¹⁷ and surveys^{18,19} are available. In this paper, new optical SPRSs are introduced. These recognizers use the structural information about a pattern. Using the pattern's structural information reduces the possibility of misclassification, even in the presence of image noise or occlusion, leading to a more robust optical filter characteristic.

The paper is organized as follows:

Background information on a SPRS is given in Sec. II. In Sec. III, a new parallel symbolic context-free language SPR algorithm is presented and, using Roman letters as an example, illustrated. In Sec. IV, a number of optical syntactic pattern recognition implementation architectures are described. Finally, Sec. V is a summary.

II. Background

In this section, after a brief review of some basic CFL definitions, the serial Earley parsing algorithm is presented. Some examples of its application to noise-free syntactic pattern recognition are also given. In a second approach, the pattern is encoded as a relational tree. A graph encoding approach is also presented.

Definition 1. A context-free language is a four-tuple $G = (V, V_T, P, S)$, where V is a finite set called the

vocabulary, V_T is the set of terminal symbols, and $V_N = V - V_T$ is the set of nonterminals. $V^*(V_T^*)$ is the set of all finite length strings over $V(V_T)$. P is a finite subset of $V_N \times V^*$ called the rules of G ; elements of P are usually written in the form $A \rightarrow \alpha$, where $A \in V_N$ and $\alpha \in V^*$. $S \in V_N$ is called the start symbol. The empty string is denoted by λ . The relation \Rightarrow on $V^* \times V^*$ is defined by $\alpha \Rightarrow \beta$ for $\alpha, \beta \in V^*$ if there are $\alpha', \alpha'', \gamma \in V^*$, and $A \in V_N$ so that $\alpha = \alpha'A\alpha''$, $\beta = \alpha'\gamma\alpha''$, and $A \rightarrow \gamma \in P$. \Rightarrow^* is a reflexive closure of \Rightarrow . The language generated by G is $L(G) = \{x \in V_T^* | S \Rightarrow^* x\}$.

Convention. Throughout the rest of this paper Roman capitals from the front of the alphabet (A, B, C, ...) denote elements of V_N . Roman small letters (a, b, c, ...) denote elements of V_T . Small Greek letters ($\alpha, \beta, \gamma, \dots$) are elements of V^* . Rules in P are designated in the form $A \rightarrow \alpha$. It is also assumed there is an integer $n \geq 0$, and a sequence of $a_1 a_2 a_3 \dots a_n$, where $1 \leq i \leq n$, $a_i \in V_T$.

Definition 2. A CFL recognizer is an algorithm which decides whether a string $a_1 a_2 \dots a_n$ is in $L(G)$.

Definition 3. Let \cdot be a symbol not found in V . For any rule $A \rightarrow \alpha\beta$ in P , we call $A \rightarrow \alpha \cdot \beta$ a dotted rule. The dot is used as a marker to indicate that the α part of the rule has been found consistent with the input; β remains to be considered.

Definition 4. Let Q be a set of dotted rules and RCV :

$$Q \cdot R = \{A \rightarrow \alpha U \cdot \beta \cdot \gamma | A \rightarrow \alpha \cdot U \cdot \beta \gamma \in Q, \beta \Rightarrow^* \lambda \text{ and } U \in R\},$$

$$Q \circ R = \{A \rightarrow \alpha U \cdot \beta \cdot \gamma | A \rightarrow \alpha \cdot U \cdot \beta \gamma \in Q, \beta \Rightarrow^* \lambda,$$

and there is some $U' \in R$ so that $U \Rightarrow^* U'\}$.

Definition 5. Let Q and R be sets of dotted rules:

$$Q \cdot R = \{A \rightarrow \alpha U \cdot \beta \cdot \gamma | A \rightarrow \alpha \cdot U \cdot \beta \gamma \in Q, \beta \Rightarrow^* \lambda \text{ and } U \rightarrow \delta \cdot \in R\},$$

$$Q \circ R = \{A \rightarrow \alpha U \cdot \beta \cdot \gamma | A \rightarrow \alpha \cdot U \cdot \beta \gamma \in Q, \beta \Rightarrow^* \lambda,$$

and there is some $U' \rightarrow \delta \cdot$ in R so that $U \Rightarrow^* U'\}$.

Definition 6. Let RCV :

$$\text{PREDICT}(R) = \{C \rightarrow \gamma \delta | C \rightarrow \gamma \delta \text{ in } P \text{ and } \gamma \Rightarrow^* \lambda,$$

$$\text{and } B \Rightarrow^* C \text{ for some } B \text{ in } R \text{ and some } \eta\}.$$

Definition 7. Let R be a set of dotted rules:

$$\text{PREDICT}(R) = \text{PREDICT}(\{B | A \rightarrow \alpha \cdot B \cdot \beta \text{ is in } R\}).$$

In terms of the dotted rule \odot and \cdot operators, Graham *et al.*⁹ rewrote the Earley algorithm. We denote it as the GHR algorithm. The GHR constructs a recognition matrix $T = \{t(i, j)\}$. All elements of T are sets of dotted rules. If $S\alpha \cdot \in t(0, n)$, we can say that the input string has been correctly recognized.

GHR algorithm

$$t(0, 0) = \text{PREDICT}(\{S\});$$

for $j = 1$ to n do

begin [build columns j , given columns $0, \dots, j-1$].

[scanner:]

for $0 \leq i \leq j-1$ do

$$t_{ij} = t_{i,j-1} \cdot |a_j|;$$

[completer:]

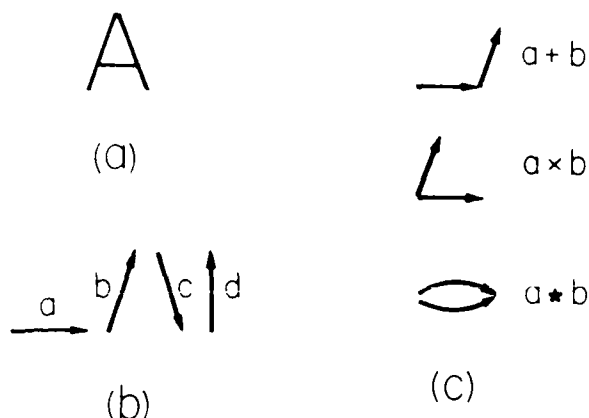


Fig. 1. Syntactic pattern example using the Roman letters A and F as input: (a) the letters; (b) the chain code primitives; (c) the interconnection operators.

```

for k := j - 1 down to 0 do
begin
  tk,k = tk,k * tk,k;
  for i := k - 1 down to 0 do
    tk,i = tk,i U tk,k * tk,i;
  begin
    tk,k = tk,k * tk,k;
    for i := k - 1 down to 0 do
      tk,i = tk,i U tk,k * tk,i;
    end;
  end;
end;

```

[Predictor:]

```

tk,j = PREDICT ( U(m,j-1) tk,j );
end;

```

Theorem 1.⁹ After executing GHR algorithm $A \rightarrow \alpha$, $\beta \in t_{k,j}$ iff $S \Rightarrow a_1, \dots, a_i A \gamma$ for some $\gamma \in V^*$ and $\alpha \Rightarrow a_{i+1}, \dots, a_j$.

For example, Fig. 1(a) depicts the character A, while Fig. 1(b) describes the pattern primitives. If we follow the primitive interconnection operators¹ shown in Fig. 1(c), the context-free grammar G (Table I) can generate the characters A and F. The characters A and F can be described as the strings

$$(b + ((b + c) * a) + c)) \text{ and } (d + (a \times (d + a))),$$

respectively. The upper triangular recognition matrix T using the GHR algorithm for this example is shown in Table II.

A second approach represents the known and unknown patterns in the form of a string. The string in this case is not just concatenation of symbols, but it represents relations, important relations, between subpatterns coded and stored in the string. For our particular example, we consider two binary relations (Fig. 2). The structure of character A [see Fig. 1(a)] in character components is as follows:

(1) Contour tracing—to determine the boundary of each character component.

(2) Search—to find some segment to be used as a start point. (It is unimportant which particular segment of a component is found.)

(3) Crawl—to crawl along a segment in a given direction, halting when an intersection or tip of a seg-

Table I. Context-Free Grammar for the Roman Letters A and F

Let $G = (V_N, V_T, P, S)$,
 where $V_N = \{S, S'\}$ and
 $V_T = \{a, b, c, d, t, +, \times, *, \cdot\}$ and
 P : (1) $S \rightarrow a$ (6) $S' \rightarrow +$
 (2) $S \rightarrow b$ (7) $S' \rightarrow \cdot$
 (3) $S \rightarrow c$ (8) $S' \rightarrow \times$
 (4) $S \rightarrow d$ (9) $S' \rightarrow *$
 (5) $S \rightarrow (SS'S)$

Table II. Nonzero Elements of the Symbolic Recognition Matrix T Computed Using the GHR Algorithm

$t_{0,0} = t_{1,1} = t_{3,3} = t_{4,4} = t_{5,5} = t_{6,6} =$
 $t_{8,8} = t_{11,11} = t_{14,14} = \{S \rightarrow a\}$
 $= \{S \rightarrow b\}$
 $= \{S \rightarrow c\}$
 $= \{S \rightarrow d\}$
 $= \{S \rightarrow (SS'S)\}$
 $t_{2,2} = t_{7,7} = t_{10,10} = t_{13,13} = \{S' \rightarrow +\}$
 $= \{S' \rightarrow \cdot\}$
 $= \{S' \rightarrow \times\}$
 $= \{S' \rightarrow *\}$
 $t_{0,1} = t_{1,2} = t_{4,5} = t_{5,6} = \{S \rightarrow (SS'S)\}$
 $t_{0,2} = t_{5,7} = t_{4,10} = t_{13,13} = \{S' \rightarrow (SS'S)\}$
 $t_{1,2} = t_{6,7} = \{S \rightarrow b\}$
 $t_{0,3} = t_{5,8} = t_{4,11} = t_{13,14} = \{S' \rightarrow (SS'S)\}$
 $t_{2,3} = t_{7,8} = t_{10,11} = \{S' \rightarrow +\}$
 $t_{8,9} = t_{14,15} = \{S \rightarrow c\}$
 $t_{5,9} = t_{4,12} = t_{13,15} = t_{0,16} = \{S \rightarrow (SS'S)\}$
 $t_{8,10} = t_{4,13} = t_{13,16} = t_{0,17} = \{S \rightarrow (SS'S)\}$
 $t_{10,11} = \{S' \rightarrow \cdot\}$
 $t_{11,12} = \{S' \rightarrow a\}$

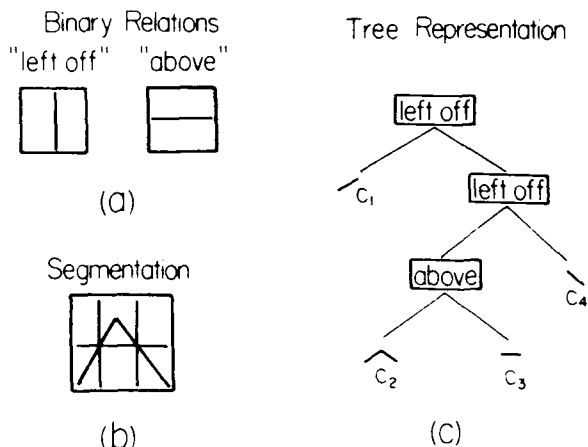


Fig. 2. Graph of the Roman letter A: (a) the binary relation operators; (b) the segmentation scheme; (c) relational tree representation of the letter A.

ment (of a component) is encountered. The octal code is used to represent the direction of a segment.

(4) Build—to construct a graph to represent the component in terms of segments.

A numeric code is generated for each component in a character. The code is generated from its graph representation. The generating algorithm is guided by the following set of rules:

(1) Start at the node in the upper left-hand corner of the graph. Exit by the branch with the lowest-

valued label. Mark the exiting branch to indicate it has been taken and write down the branch label.

(2) On entering a node, check to see if it is being visited for the first time. If it is being visited for the first time, mark this node.

(3) On leaving a node, choose an unmarked branch with the lowest-valued label among available unmarked branches. The code consists of the branch labels in the graph written down in the order in which they are encountered.

Once the subpatterns are analyzed and recognized, the character made of these subpatterns and relations between them can be identified. Each character is represented by a sequence of $n + 1$ codes, where n is the number of components in the character, i.e.,

$$C_0, C_1, C_2, \dots, C_n,$$

where C_i is the code generated from the relational tree representation of the character. It is generated by traveling around the tree in a counterclockwise direction starting from the root node of the tree and picking up nodes and leaves (terminals) the first time as they are encountered.

The subpattern analysis and recognition procedures are illustrated using the example of the character A (Fig. 3). The C_0 for character A is $C_0 = 101200$. There are four components of character A shown in the tree representation [Fig. 2(c)]. The steps to represent a component by a code are illustrated specifically for the components C_2 [Fig. 3(b) and (c)]. The code representations for other components are as follows:

$$C_1 = 73, C_3 = 26, \text{ and } C_4 = 15.$$

The code representation of character A is

$$C_0, C_1, C_2, C_3, C_4 = (101200) (73) (7315) (26) (15).$$

In a similar manner, the code representation of character F can also be obtained. Recognition of a character is based on the matching between the sequence of $n + 1$ codes representing the unknown character and the codes specifying different known patterns. For this specific example, let us suppose that an unknown pattern is to be identified. The unknown pattern can also be represented by a sequence of $n + 1$ codes using the method described above. This new sequence is matched against the reference codes representing A and F. If a match is not found, we conclude that the unknown pattern is neither the character A nor the character F.

III. Parallel CFL Recognition

The GHR is a sequential algorithm. Its order of computation demands that no elements of column $j + 1$ be processed until $t_{j,j}$ is processed and hence until all elements of column j are processed. This restriction precludes a parallel implementation of a GHR algorithm. However, by replacing $t_{j,j}$ with a constant set of dotted rules $C = \text{PREDICT}(V_N)$, one may remove the predictor operation and still preserve the correctness of the recognition algorithm. After replacing $t_{j,j}$ with

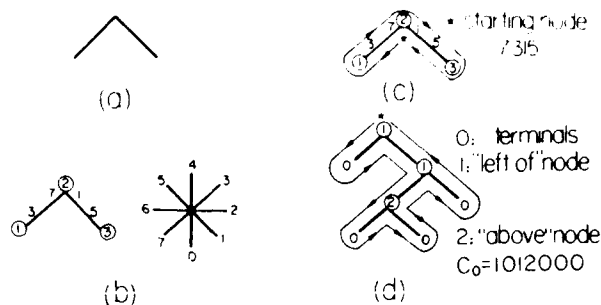


Fig. 3. Encoding of the graph of Fig. 2: (a) a component of A; (b) a graph representation of (a) using the octal chain code; (c) the code representation of the graph of Fig. 2(c).

C_i , there is no need to calculate the diagonal elements of recognition matrix T . Based on this unrestricted computation, the following parallel version of Earley's algorithm is proposed.²⁰

Parallel Earley algorithm

for $i = 1$ to n do in parallel

$$t(i-1, i) = C \cdot a_i;$$

$$t(i-1, i) = C \odot t(i-1, i);$$

for $j = 2$ to n do

for $i = 0$ to $n - j$ do in parallel

begin

$$t(i, i+j) = t(i, i+j-1) \cdot a_{i+j};$$

for $k = j - 1$ to 1 do in parallel

$$t(i, i+j) = t(i, i+j) \cup t(i, i+k) \cdot t(i+k, i+j);$$

$$t(i, i+j) = C \odot t(i, i+j);$$

end;

A computation diagram comparing the two serial-parallel parse algorithms is shown in Fig. 4.

The proposed parallel version of Earley algorithm is used to recognize the character A [Fig. 1(a), assumed to be an unknown character]. First, the unknown character of Fig. 1(a) is represented as a string. We have used the primitives of Fig. 1(b) and the primitive interconnection operators of Fig. 1(c) to represent the unknown character as a string. The grammar G generating characters A and F is given in Table I. Our aim is to decide if this unknown string belongs to the language generated by the grammar G . The upper triangular matrix T generated by the parallel algorithm is given in Table III. The element $t_{0,17}$ contains a dotted rule of the form $[S \rightarrow (SS'S)]$. The presence of such a dotted rule indicates that the input string is in $L(G)$. Therefore, the unknown character of Fig. 1(a) is classified to a set which consists of characters A and F.

IV. Optical Syntactic Pattern Recognizer

In this section, a number of parallel optical SPR are described. These systems are based on the proper optical encoding of the pattern's structural information. Three optical SPRs, the structural-information encoded matched filter, an associative memory SPR, and optical symbolic processing SPR architectures, are briefly described.

An SPRS always contains a preprocessing stage where the physical input pattern is smoothed, en-

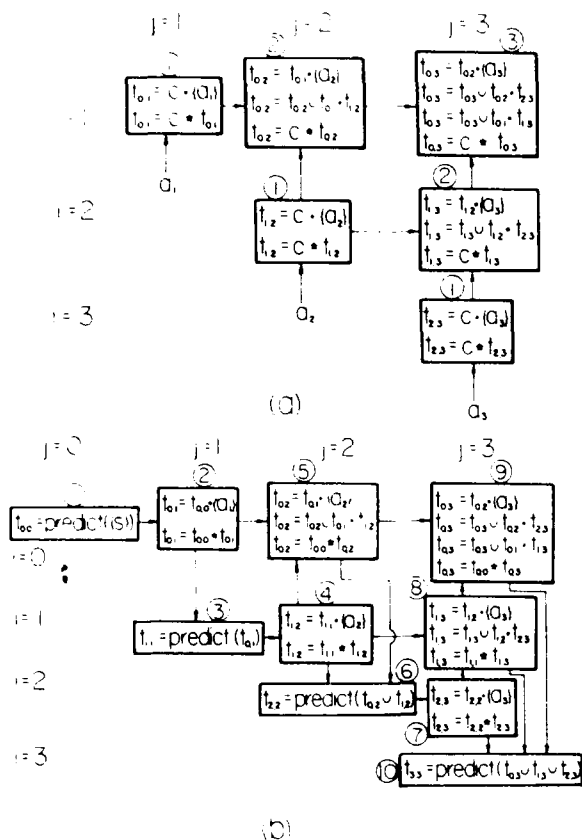


Fig. 4. Computation diagram for a three-symbol input string using the two parsing algorithms. The order of the computation is indicated above the boxes: (a) the GHR parse algorithm; (b) the parallel parse algorithm.

hanced, and encoded. In an optical SPRS, the encoding stage uses both data compressive and optical encoding methodologies. If the pattern encoding stage uses its structural information, for example, with the graph-theoretic encoding approach illustrated in Fig. 2, this new pattern may be considered as a transformed version of the original physical pattern. This transformed with its structural information encoded image may be stored in a structural-information encoded matched filter. To recognize this pattern, the incoming image when it is properly encoded must be correlated with the stored reference patterns in either a Fourier spatial frequency or a correlation (template) matching plane. Once a strong correlation is achieved, the proper physical image may be reconstructed.

An alternative approach is to use an optical associative memory filter (AMF) formalism²¹⁻²⁴ to build an optical SPRS. The idea behind the AMF is that, instead of using a SPR computing algorithm, the allowable set of input-output pattern sets is memorized. The optical AMF can consist of either a linear, a correlative or its equivalent the holographic, or a nonlinear, a threshold, or its equivalent a neural, type of network element. Using a proper and desired training set, it

Table III. Nonzero Elements of the Symbolic Recognition Matrix T Computed Using the Parallel Parsing Algorithm

$t_{0,1} = t_{1,4} = t_{4,5} = t_{5,6} = [S \rightarrow (SS'S'S)]$
$t_{0,2} = t_{1,7} = t_{4,10} = t_{5,11} = [S \rightarrow (SS'S'S)]$
$t_{1,2} = t_{6,7} = [S \rightarrow b]$
$t_{0,3} = t_{5,8} = t_{4,11} = t_{5,14} = [S \rightarrow (SS'S'S)]$
$t_{2,3} = t_{7,8} = t_{13,14} = [S' \rightarrow +]$
$t_{8,9} = t_{14,15} = [S \rightarrow c]$
$t_{5,9} = t_{4,12} = t_{3,15} = t_{0,16} = [S \rightarrow (SS'S'S)]$
$t_{5,10} = t_{4,13} = t_{3,16} = t_{0,17} = [S \rightarrow (SS'S'S)]$
$t_{10,11} = [S \rightarrow a]$
$t_{11,12} = [S \rightarrow a]$

can be trained to recognize appropriate inputs. For a given input, the output can be precomputed using any available SPR algorithm. The AMF output can be either a tag, a number in a catalog of acceptable patterns, or the physical pattern itself. While the physical image AMF itself, without any structural information encoding, can be quite resistant²⁵ to noise and some topological changes, the addition of structural information redundancies should further enhance this trend.

Finally, a third approach, the use of an optical SPR symbolic computing approach, is described. Recently, a number of methodologies have been suggested for the realization of an optical inference engine.²⁶⁻³⁰ In another approach, an optical symbolic substitution method was designed as a means for eliminating bottlenecks in conventional electronic systems.^{31,32} With the optical symbolic substitution approach, the addition of binary and trinary³³ digits can be affected. In our approach, rather than encoding the finite numbers, the grammar symbols and their symbolic relations are optically encoded. In Fig. 5, as an example, the optical symbolic substitution approach for the Roman letters of Fig. 1 is indicated. In Fig. 5(a), a possible optical symbolic encoding scheme for the symbols in the grammar $L(G)$ is indicated. The binary masks can either be long 1-D slices or 2-D areas. In Fig. 5(b), the symbolic substitution relations corresponding to the grammar of Table I is indicated. These symbolic substitutions comprise the elements of a parse table. As an example, in Fig. 5(c), a fragment of the symbolic substitutions relations of Table II is shown. By performing sequentially according to a prescribed parse algorithm, the necessary symbolic substitutions and together with an optical matched filter which serves to detect the successful completion of the parse, an optical symbolic SPR can be constructed. While optical syntactic matched filters or associative memory filters are conceptually straightforward, in practice, for realistic patterns, they can be quite large. The method of symbolic substitution has the advantage that, even for large complex patterns, the number of grammar symbols and the inference relations defining the grammar can be kept relatively small.³⁴ Using a rapid optical symbolic substitution device, a compact optical SPRS can be constructed.

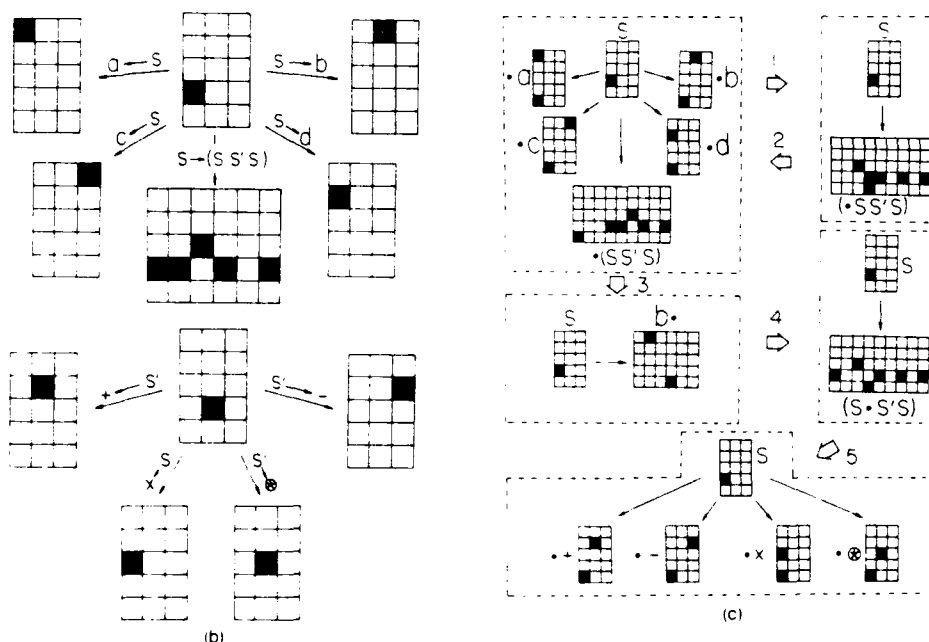
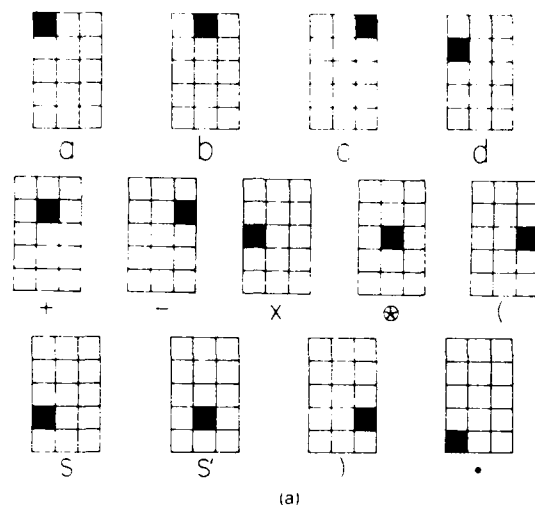


Fig. 5. Optical symbolic substitution encoding for the Roman letters A and F: (a) optical encoding of the symbols in $L(G)$; (b) optical symbolic substitution encoding of the grammar $L(G)$; (c) optical substitution scheme for a fragment of the parse Table II.

V. Summary

In this paper, the use of parallel optical syntactic pattern recognizers was discussed. These recognizers utilize the pattern's structural information. Because of this additional information, robust pattern recognition is possible. After describing some of the presently available SPR approaches, a new parallel parsing algorithm was presented and illustrated. This algorithm requires fewer parsing steps for recognition. A number of optical approaches for SPRS were de-

scribed. An optical structural matched filter and optical associative memory filter approaches were described. The use of the method of symbolic substitution to generate optical symbolic parse tables has also been indicated. Such symbolic substitutions find use in other artificial intelligence modeling applications.

This work is supported in part by a grant from the Air Force Office of Scientific Research.

References

1. A. C. Shaw, "A Formal Picture Description Scheme as a Basis for Picture Processing Systems," *Int. Control* **14**, (1969).
2. K. S. Fu, *Syntactic Pattern Recognition and Applications* (Prentice Hall, Englewood Cliffs, NJ, 1982).
3. K. Fukunaga, *Introduction to Statistical Pattern Recognition* (Academic, New York, 1972).
4. H. S. Don and K. S. Fu, "A Syntactic Method for Image Segmentation and Object Recognition," *Pattern Recognition* **18**, 73 (1985).
5. K. S. Fu, Ed., *Syntactic Pattern Recognition and Applications* (Springer-Verlag, New York, 1977).
6. K. S. Fu, "Syntactic Pattern Recognition and Applications," in *Proceedings, COMPCON Spring*, San Francisco, 26-28 Feb. 1980.
7. K. S. Fu, Ed., *Applications of Pattern Recognition* (CRC Press, Boca Raton, FL, 1981).
8. J. Earley, "An Efficient Context-Free Parsing Algorithm," *Commun. ACM* **13**, 94 (1970).
9. S. L. Graham, M. A. Harrison, and W. L. Ruzzo, "On-Line Context Free Language Recognition in less than Cubic Time," in *Proceedings, Eighth Annual ACM Symposium on Theory of Computing* (1976), pp. 112-120.
10. R. Weicker, "General Context-Free Language Recognition by a RAM with Uniform Cost Criterion in Time $n^2 \log n$," *Tech. Rep. 182*, Computer Science Department, Pennsylvania State U. (1976).
11. A. B. Vanderlugt, "Signal Detection by Complex Spatial Filtering," *IEEE Trans. Inf. Theory* **IT-10**, 2 (1964).
12. H. H. Arsenault, Y. N. Hsu, and K. Chalasinska-Macukow, "Rotation-Invariant Pattern Recognition," *Opt. Eng.* **23**, 705 (1984).
13. G. Eichmann and N. M. Marinovic, "Scale-Invariant Wigner Distribution and Ambiguity Functions," *Proc. Soc. Photo-Opt. Instrum. Eng.* **519**, 17 (1985).
14. J. R. Leger and S. H. Lee, "Hybrid Optical Processor for Pattern Recognition and Classification Using a Generalized Set of Pattern Functions," *Appl. Opt.* **21**, 275 (1982).
15. Z.-H. Gu and S. H. Lee, "Recognition of Images of Markov-1 Model by Least-Squares Linear Mapping Technique," *Appl. Opt.* **23**, 822 (1984).
16. H. J. Caulfield and M. H. Weinberg, "Computer Recognition of 2-D Patterns Using Generalized Matched Filters," *Appl. Opt.* **21**, 1699 (1982).
17. J. L. Horner, Ed., Special Issue of Optical Engineering on Optical Pattern Recognition, *Opt. Eng.* **23**, No. 6 (1984).
18. D. Casasent, "Coherent Optical Pattern Recognition," *Proc. IEEE* **67**, 813 (1979).
19. D. Casasent, "Pattern Recognition: A Review," *IEEE Spectrum* **18**, 28 (1981).
20. S. Basu, "A Parallel Syntactic Pattern Recognition Algorithm," in preparation.
21. D. Psaltis and N. Farhat, "Optical Image Processing Based on an Associative-Memory Model for Neural Nets with Thresholding and Feedback," *Opt. Lett.* **10**, 98 (1985).
22. N. H. Farhat, D. Psaltis, A. Prata, and A. E. Peak, "Optical Implementation of the Hopfield Model," *Appl. Opt.* **24**, 1469 (1985).
23. B. H. Soffer, G. J. Dunning, Y. Owechko, and E. Marom, "Associative Holographic Memory with Feedback Using Phase-Conjugate Mirrors," *Opt. Lett.* **11**, 118 (1986).
24. R. A. Athale, H. H. Szu, and C. B. Friedlander, "Optical Implementation of Associative Memory with Controlled Nonlinearity in the Correlation Domain," *Opt. Lett.* **11**, 482 (1986).
25. G. Eichmann, M. Jankowski, and M. Stojancic, "Shape Description with an Associative Memory," *Proc. Soc. Photo-Opt. Instrum. Eng.* **638** (1986).
26. G. Eichmann and H. J. Caulfield, "Optical Learning (Inference) Machines," *Appl. Opt.* **24**, 2051 (1985).
27. C. Warde and J. Kottas, "Hybrid Optical Inference Machines: Architectural Considerations," *Appl. Opt.* **25**, 940 (1986).
28. D. Casasent, "Optical AI Symbolic Correlators: Architecture and Filter Consideration," *Proc. Soc. Photo-Opt. Instrum. Eng.* **625**, 220 (1986).
29. D. Casasent and A. J. Lee, "A Feature-Space Ruled-Based Optical Relational Graph Processor," *Proc. Soc. Photo-Opt. Instrum. Eng.* **625**, 234 (1986).
30. A. D. Fisher, R. C. Fukuda, and J. N. Lee, "Implementations of Adaptive Optical Computing Elements," *Proc. Soc. Photo-Opt. Instrum. Eng.* **625**, 196 (1986).
31. A. Huang, "Parallel Algorithms for Optical Computers," in *Proceedings, IEEE Tenth International Optical Computing Conference*, IEEE Catalog 83CH1880-4 (1983), p. 13.
32. K.-H. Brenner and A. Huang, "An Optical Processor Based on Symbolic Substitution," in *Technical Digest, Topical Meeting on Optical Computing* (Optical Society of America, Washington, DC, 1985), paper WA4.
33. R. P. Bocker, B. L. Drake, M. E. Lasher, and T. B. Henderson, "Modified Signed-Digit Addition and Subtraction Using Optical Symbolic Substitution," *Appl. Opt.* **25**, 2456 (1986).
34. S. Basu and G. Eichmann, "Optical Error-Correcting Syntactic Pattern Recognizers," to appear in *Proc. Soc. Photo-Opt. Instrum. Eng.* **752**, 13 (1987).

Superresolving signal and image restoration using a linear associative memory

George Eichmann and M. Stojancic

The long standing problem of the superresolving reconstruction (restoration) of an object of known finite spatial extent from a noisy linearly degraded image is considered. The resolution of two-point sources (objects) spaced less than one Rayleigh distance apart is an ill-posed problem. To determine a superresolving inverse of an ill-conditioned linear degradation operator with a known set of input-output training signals, a linear associative memory (LAM) technique is employed. By limiting the set of reconstructable signals, an exceptionally robust inverse filter has been obtained. This filter is based on a new constrained LAM matrix operator technique. Superresolving restoration of 1-D and 2-D two-point sources as well as some typical edge-type signals in the presence of considerable measurement noise is demonstrated.

1. Introduction

An important problem in signal and image processing is the recovery or restoration of a signal or image that has been distorted by a linear degradation operator. This problem is common to different application areas such as diffraction optics, radio astronomy, and electron microscopy. It can be stated, in general, as the problem of reconstruction of a Fourier transform pair $|f, F|$ from partial data on either/or both domains (spatial and/or spatial frequency).¹ The extrapolation of bandlimited signals has been addressed by several authors.²⁻⁷ A number of iterative extrapolation algorithms, for example, Gerschberg-Saxton,² Papoulis,³ and Youla,⁴ and also direct algorithms, such as Cadzow⁵ and Sabry and Steenart,⁶ have been proposed. An extended analysis and generalization of constrained iterative algorithms are given in Ref. 7. When the measurement noise is negligible and the degradation process is well-behaved,^{8,9} direct spatial frequency domain inversion of the degradation process is possible. In the presence of noise, a method such as Wiener filtering¹⁰ is more appropriate. However, in many image restoration problems, the measurement error is greatly amplified so that the error dominates the estimate. Image restoration methods which im-

pose either smoothness or positivity constraints, regularization techniques, or techniques that employ some solution criteria have also been widely analyzed.¹¹⁻¹⁹

The effect of the finite aperture of a system is characterized by an ideal low-pass filter (ILPF) with its cutoff frequency u_c . If a signal $f(x)$ has been passed through an ILPF, the corresponding output signal will not contain high-frequency components beyond $u = u_c$. In general, the original signal $f(x)$ cannot be reconstructed from the given ILPF output. However, if the input is space-limited, i.e., $f(x) = 0$ outside an interval $L = l_2 - l_1$, it can be recovered from its integral equation form²⁰

$$g(x) = \int_{l_1}^{l_2} f(t) \frac{\sin[u_c(x-t)]}{u_c(x-t)} dt. \quad (1)$$

The properties of the eigenvalues and eigenfunctions of the integral Eq. (1) have been thoroughly studied.²⁰ The eigenfunctions $q_n(x)$ are the prolate spheroidal wave functions with the space-bandwidth product (SBP) as their parameter and with associated eigenvalues λ_n , where $1 > \lambda_1 > \dots > \lambda_n > \dots > 0$, which for $n > \text{SBP}$ decrease rapidly to zero. The input can be expanded as

$$f(x) = \sum_{n=0}^{\infty} \lambda_n^{-1} a_n q_n(x), \quad (2)$$

where

$$a_n = \int_{l_1}^{l_2} g(x) q_n(x) dx. \quad (3)$$

From Eq. (2) it is clear that since λ_n^{-1} tends to infinity for large n , slight measurement errors in the output can lead to large changes in the restored object function.

The authors are with City University of New York, City College, Department of Electrical Engineering, New York, New York 10031.
 Received 7 October 1986.
 0003-6935/87/101911-08\$02.00/0.
 © 1987 Optical Society of America.

Both in optics and spectroscopy, due to the finite aperture of physical imaging systems, there is a diffraction effect-induced resolution limit. In principle, because $f(x)$ has finite support, its spectrum is an entire function and can be analytically continued beyond its cutoff frequency. Continuation of the spectra in the spatial frequency domain provides an increase of resolution in the spatial domain. A frequently used resolution criterion is the so-called Rayleigh distance.⁹ For a given ILPF, with a two-sided spatial bandwidth W , this is the distance between two ideal impulses S so that the maximum of the impulse response (IR) of the first impulse overlaps the first zero of the IR of the second impulse. This condition translates as the value of unity for the SBP. If SBP is below unity, the so-called uncertainty principle,²¹ well known in signal processing, holds. An increase of resolution beyond the Rayleigh limit is called superresolving. Few signal and image restoration algorithms exhibit superresolving properties substantially below the Rayleigh limit. While most linear nonconstrained deterministic techniques achieve in the absence of noise a resolution of the order of half of a Rayleigh length,⁸ in general, statistical methods are not superresolving. On the other hand, to provide numerical stability, most nonlinear direct and/or iterative methods use various forms of regularization, a process that limits their superresolving properties.

Direct discrete solution of the integral Eq. (4) can be obtained employing some numerical quadrature rule.²² Assuming that the number of samples of a measured (degraded) sampled object vector \mathbf{g} is equal to the number of samples of the sampled object vector \mathbf{f} , we have

$$\mathbf{g} = \mathbf{H}\mathbf{f}, \quad (4)$$

where \mathbf{H} is an $N \times N$ degradation matrix operator obtained from the quadrature rule. The solution of \mathbf{f} in Eq. (4) requires a matrix inversion. As in the continuous case, Eq. (1), because the higher-order eigenvalues of \mathbf{H} tend rapidly to zero, \mathbf{H} is a nearly singular matrix. Thus the inverse of \mathbf{H} does not exist, or, if it exists, during the inversion process the higher-spatial-frequency components of the signal are greatly amplified causing numerical instability. If the degraded image \mathbf{g} is noisy, even if \mathbf{H}^{-1} (where the superscript stands for the inverse) can be properly approximated, the result of restoration may differ radically from the true solution.

To determine an approximate superresolving inverse \mathbf{H}^{-1} , in this work, a set $\{(\mathbf{f}_k, \mathbf{g}_k)\}$, $k = 1, 2, \dots, N$ of known input/output training signals and a linear associative memory (LAM) technique is employed. The LAM relates the two given vectors \mathbf{f} and \mathbf{g} by the matrix equation²³

$$\mathbf{f} = \mathbf{M}\mathbf{g}, \quad (5)$$

where \mathbf{M} is an unknown matrix. In this terminology, the vectors (\mathbf{f}, \mathbf{g}) are identified as the data and key vectors, respectively. The LAM matrix \mathbf{M} represents a content addressable memory (CAM) that allows by specifying the proper key the data recall. If the key

and data are identical, the recall is termed autoassociative. Otherwise it is a heteroassociative recall.²³ To identify, or restore superresolving linearly degraded signals or images, in the presence of appreciable noise, a heteroassociative recall operation is proposed. In this model of associative memory, the k th item to be stored consists of two parts, the key vector \mathbf{g}_k and the data vector \mathbf{f}_k . The key and data vectors are encoded as column vectors in the key \mathbf{X} and the data \mathbf{Y} matrices, respectively. The number of elements in the key \mathbf{g} and data \mathbf{f} vectors need not be the same. The recall LAM matrix is $\mathbf{M} = \mathbf{Y}\mathbf{X}^+$, where $+$ denotes the Moore-Penrose generalized inverse.²⁴ The recall operation is performed by multiplying an input key vector \mathbf{g}' (which can be a noisy or corrupted vector) by \mathbf{M} to yield the desired data vector \mathbf{f}' . This result is the minimum norm least-squares approximation to data vector \mathbf{f} , paired to the key vector \mathbf{g} , that is closest in the sense of least squares to the input vector \mathbf{g}' . In the case of heteroassociative recall it has been found²⁵ that for certain structures of the key or input matrix \mathbf{X} , while the conditions for noiseless recall may be satisfied, the LAM can be sensitive to input noise. As the determinant of $(\mathbf{X}^T\mathbf{X})$ approaches zero, in spite of the linear independence of the columns of \mathbf{X} , the elements of the LAM matrix \mathbf{M} may become unacceptably large. A very similar problem arises in the application of LAM to superresolving signal restoration, when, due to small SBP, the key matrix \mathbf{X} contains a large number of similar columns. To achieve superresolution, in this paper, a constrained optimization of the dynamic range of the LAM matrix \mathbf{M} elements with respect to the particular data-key set is proposed. The LAM matrix \mathbf{M} is determined by a new constrained recursive training algorithm with a set of known input/output sampled vector pairs $\{(\mathbf{f}_k, \mathbf{g}_k)\}$, $k = 1, 2, 3, \dots, N$. The training set represents a known data-key set. Using this modified LAM matrix \mathbf{M} , it will be shown that it is possible to attain superresolution far below the Rayleigh limit even in the presence of large measurement noise.

This paper is organized as follows: In Sec. II, after presenting the recursive matrix training algorithm, the application of a constrained LAM matrix to superresolving restoration is described. In Sec. III, the superresolving results of a two-point noisy object function are presented. Both 1-D signal and 2-D image superresolving restoration for a different level of noise and space-bandwidth products are demonstrated. Also, the results of superresolving restoration of particular test images, such as a triangle, trapezoid, and Gaussian pulse of different width and slope edges, are considered. Superresolving restoration in the presence of considerable additive noise and different space-bandwidth products is demonstrated. Finally, in Sec. IV, summary and conclusions are presented.

II. Superresolving Signal and Image Restoration (SIR) Using the LAM Technique

In this section, the constrained iterative training process for determination of the LAM matrix \mathbf{M} and its

application to superresolving signal and image restoration is described.

In the linearly degraded image restoration problem, Eq. (4) relates the object \mathbf{f} to the degraded image \mathbf{g} by a linear degradation operator \mathbf{H} . However, in the presence of noise Eq. (4) is amended:

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n}, \quad (6)$$

where \mathbf{n} is assumed to be an uncorrelated additive noise vector with zero-mean and identical variance elements. It can be shown²⁴ that a minimum least-squares norm estimate of \mathbf{f} is

$$\mathbf{f} = \mathbf{H}^+ \mathbf{g}, \quad (7)$$

where \mathbf{H}^+ is the Moore-Penrose generalized (pseudo) inverse.²⁴ Based on the given observation (measurement) vector \mathbf{g} , this estimate for the vector \mathbf{f} coincides with the best linear unbiased estimate (BLUE).²⁴ The pseudo inverse \mathbf{H}^+ can in general be determined by a singular value decomposition (SVD).²⁶ The SVD method clearly indicates that the eigenvalues H for a discrete ILPF as a function of the space-bandwidth product change abruptly from approximately unity to nearly zero. This fact introduces numerical instability in the determination of the inverse of \mathbf{H} . To deal with this problem, a number of direct or iterative regularized algorithms exist.¹⁵⁻¹⁹ In general, these regularization algorithms tend to attenuate those signal components that correspond to the high-order eigenvalues of \mathbf{H} at the expense of superresolution, where precisely these high-order eigenvalues are needed.²⁷

To introduce the constrained LAM, consider the given set of the estimates of the object vectors $\{\hat{\mathbf{f}}_k\}$ and related linearly degraded vectors $\{\mathbf{g}_k\}$, $k = 1, 2, \dots, N$. Each pair of vectors $(\hat{\mathbf{f}}_k, \mathbf{g}_k)$ can be related by the matrix \mathbf{M} . We form the key \mathbf{X} and data \mathbf{Y} matrices with the column vectors \mathbf{g}_k and $\hat{\mathbf{f}}_k$, respectively, as

$$\begin{aligned} \mathbf{X} &= [\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_N], \\ \mathbf{Y} &= [\hat{\mathbf{f}}_1, \hat{\mathbf{f}}_2, \dots, \hat{\mathbf{f}}_N]. \end{aligned} \quad (8)$$

From Eq. (7), a matrix equation is obtained:

$$\mathbf{Y} = \mathbf{H}^+ \mathbf{X}, \quad (9)$$

If we identify $\mathbf{H}^+ = \mathbf{M}$ and form the \mathbf{Y} matrix from the set of original object vectors, Eq. (9) is

$$\mathbf{Y} = \mathbf{M}\mathbf{X}, \quad (10)$$

where the least-squares sense optimum solution for \mathbf{M} is again

$$\mathbf{M} = \mathbf{Y}\mathbf{X}^+, \quad (11)$$

There are a number of methods to evaluate the pseudoinverse \mathbf{X}^+ matrix.^{24,26,28-30} However, for most of them, because of ill-conditioning of the \mathbf{X} matrix for small SBP, some additional corrections are required. Here to evaluate the \mathbf{M} matrix a new constrained recursive gradient projection algorithm that is based on a theorem of Greville²⁸ on pseudo inverses of partitioned matrices (see Kohonen²³) is used. This new algorithm is a trade-off between numerical complexity and com-

putational speed. It also provides a mechanism for adding constraints to the \mathbf{M} matrix. A brief outline of the unconstrained recursive algorithm is as follows.

Let $\mathbf{M}_r = \mathbf{Y}_r \mathbf{X}_r^+$ be the r vector pair LAM matrix. The \mathbf{Y}_r matrix can be partitioned as $[\mathbf{Y}_{r-1}, \mathbf{f}_r]$. Applying Greville's theorem,²⁸ the \mathbf{X}_r^+ can be determined as

$$\mathbf{X}_r^+ = \begin{bmatrix} \mathbf{X}_{r-1}^+ (\mathbf{I} - \mathbf{g}_r \mathbf{c}_r^T) \\ \mathbf{c}_r^T \end{bmatrix}, \quad (12)$$

where \mathbf{I} is the $N \times N$ identity matrix, T stands for transpose, and

$$\mathbf{c}_r = \begin{cases} (\mathbf{I} - \mathbf{X}_{r-1} \mathbf{X}_{r-1}^+) \mathbf{g}_r, & \text{if the numerator is } \neq 0, \\ (\mathbf{I} - \mathbf{X}_{r-1} \mathbf{X}_{r-1}^+) \mathbf{g}_r, & \end{cases} \quad (13a)$$

$$\begin{cases} (\mathbf{X}_{r-1}^+)^T \mathbf{X}_{r-1} \mathbf{g}_r, & \\ 1 + \mathbf{X}_{r-1}^+ \mathbf{g}_r, & \text{otherwise,} \end{cases} \quad (13b)$$

where $\|\cdot\|$ represents a quadratic vector norm. Multiplying the partitioned matrix \mathbf{Y}_r by \mathbf{X}_r^+ [see Eq. (12)] yields

$$\mathbf{M}_r = \mathbf{Y}_{r-1}^+ \mathbf{X}_{r-1}^+ + (\mathbf{f}_r - \mathbf{Y}_{r-1} \mathbf{X}_{r-1}^+ \mathbf{g}_r) \mathbf{c}_r^T, \quad (14)$$

or equivalently

$$\mathbf{M}_r = \mathbf{M}_{r-1} + (\mathbf{f}_r - \mathbf{M}_{r-1} \mathbf{g}_r) \mathbf{c}_r^T, \quad (15)$$

In the case where \mathbf{g}_r is linearly independent of the previous columns of \mathbf{X} , Eq. (13a) for \mathbf{c}_r^T applies; otherwise Eq. (13b) for \mathbf{c}_r^T implies $\mathbf{M}_r = \mathbf{M}_{r-1}$.²³ Therefore, the recursive algorithm for \mathbf{M}_r given \mathbf{M}_{r-1} is

$$\mathbf{M}_r = \begin{cases} \mathbf{M}_{r-1} + (\mathbf{f}_r - \mathbf{M}_{r-1} \mathbf{g}_r) \frac{\hat{\mathbf{g}}_r^T}{\hat{\mathbf{g}}_r}, & \text{for } \hat{\mathbf{g}}_r \neq 0, \\ \mathbf{M}_{r-1} & \text{otherwise,} \end{cases} \quad (16)$$

where $\hat{\mathbf{g}}_r$ is the orthogonal projection of the vector \mathbf{g}_r to the vector subspace spanned by the column vectors of the key matrix \mathbf{X} , i.e., $(\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_{r-1})$. The orthogonal projection vector $\hat{\mathbf{g}}_r$ can be obtained either from a Gram-Schmidt orthogonalization (GSO) or a modified GSO (MGSO) procedure.²⁶ The GSO algorithm for the orthogonal projection vectors is

$$\hat{\mathbf{g}}_r = \mathbf{g}_r - \sum_{i=1}^{r-1} (\mathbf{g}_i, \hat{\mathbf{g}}_r) \hat{\mathbf{g}}_i, \quad (17)$$

where the bracket (\dots) denotes the vector inner product. Similar mechanism is also available for the MGSO procedure. In Eq. (17), the sum over i is taken for the nonzero $\|\hat{\mathbf{g}}_i\|$ only.

To start the iteration in Eq. (16), the initial LAM matrix \mathbf{M}_0 can be chosen to be either a zero or an identity matrix. However, it is well known²⁴ that for the case of a zero mean stochastic input/output vector pairs (\mathbf{f}, \mathbf{g}) the linear regression solution for \mathbf{M} is $\mathbf{M} = \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}$. This solution also coincides²³ with the BLUE solution $\mathbf{M} = \mathbf{R}_{yx} \mathbf{R}_{xx}^{-1}$, where \mathbf{R}_{yx} and \mathbf{R}_{xx} are the sample cross-correlation and autocorrelation matrices, respectively. These matrices are calculated from the column vectors of the \mathbf{Y} and \mathbf{X} matrices as

$$R_{11} = (1/N) \sum_{k=1}^N \mathbf{f}_k \mathbf{f}_k^T \quad (18)$$

$$R_{11} = (1/N) \sum_{k=1}^N \mathbf{g}_k \mathbf{g}_k^T \quad (19)$$

To improve the convergence speed and also reduce the unnecessary processing of noisy superresolving images, the covariance matrix obtained from the deterministic set of degraded training vectors $\{\mathbf{g}_k\}$

$$C_{xx} = (1/N) \sum_{k=1}^N \mathbf{g}_k \mathbf{g}_k^T \quad (20)$$

can, therefore, also serve as an initial LAM matrix estimate M_0 .

For convergence, it is desirable that X be a well-conditioned matrix. However, when the SBP is small (especially when $SBP < 1$), because the key matrix X is composed of the similar column vectors \mathbf{g}_k (degraded images of the original object \mathbf{f}_k), it is ill-conditioned. The SVD of X clearly shows that the high-order singular values δ_n as a function of SBP drop abruptly to almost zero. Thus the recursive algorithm of Eq. (16) for small SBP shows numerical instability by producing a solution of M with a very large quadratic matrix norm $\|M\|$. Furthermore, because of the large number of nearly zero vectors in the orthogonalized set $\{\mathbf{g}_k\}$, calculated by either a GSO or MGSO [see Eq. (17)], the number of algorithmic recursion steps is small. This fact suggests that a new set of projection directions $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N\}$, which may not correspond to the orthogonal directions $\{\mathbf{g}_1, \dots, \mathbf{g}_N\}$ but could provide a large number of iteration steps, should be followed. It is well known²⁶ that the condition number $c(M)$, defined as

$$c(M) = \frac{\delta_1(M)}{\delta_N(M)} \quad (21)$$

where $\delta_1(M)$ and $\delta_N(M)$ are the largest and smallest singular values of M , respectively, reflects the sensitivity of a linear system to perturbation (noise). Thus, to reduce the numerical instability associated with the large number of similar key matrix X columns and to decrease the sensitivity of the solution to noise, a new constrained LAM algorithm is introduced. With this algorithm, in several iteration steps, a new set of projection directions $\{\mathbf{p}_k\}$ is chosen so as to minimize the M matrix condition number $c(M)$.

The new set of pseudo-orthogonal vectors $\{\mathbf{p}_k\}$ is obtained as follows: a threshold parameter P is introduced, while the values of the vector elements of every term in the sum of Eq. (17) are implicitly constrained. Using the P parameter, the norm $\|\mathbf{g}_k\|$ in Eq. (17) is tested before it is added to the previous set of terms. The following decision rule is used: the $\|\mathbf{g}_k\|$ is compared to P ; if $\|\mathbf{g}_k\|$ is $< P$, the related term in the sum of Eq. (17) is discarded. Otherwise a division by $\|\mathbf{g}_k\|$ is performed, and it is accepted. Thus, for a given value of P , a new set of pseudo-orthogonal $\{\mathbf{p}_k\}$ is formed. To determine the set $\{\mathbf{p}_k\}$ that minimizes the condition

number $c(M)$, one starts with a small initial threshold parameter P_0 . For increasing values of P , new LAM matrices are generated, and their quadratic matrix norms are calculated. It has been found, however, that this procedure does not considerably affect the smallest singular value $\delta_N(M)$ so that, instead of minimizing the condition number $c(M)$, the quadratic matrix norm of M , i.e., $\delta_1(M)$, is minimized. Thus the LAM matrix M that yields the smallest quadratic matrix norm is used for signal reconstruction (associative recall).

While M is usually trained with known and desired signals, however, it can also be trained to reject known, sometimes only statistically, undesirable signals. For example, if we wish to reject pure noise signals, the training procedure is modified so as to include also a set of known noise $\{\mathbf{g}_n\}$ and desired zero vectors. (These are vectors with zero as their elements.) The new training set becomes $\{(\mathbf{f}_k, \mathbf{f}_n; \mathbf{g}_k, \mathbf{g}_n)\}$, $k = 1, \dots, M$; $n = M + 1, \dots, N$, where the pairs $(\mathbf{f}_k, \mathbf{g}_k)$ are ILPF inputs and outputs, and $(\mathbf{f}_n, \mathbf{g}_n)$ are zero and noise vectors, respectively. In this way, the M matrix can be trained to reject noise by producing a null output response. If we wish to reject either impulse noise, sinusoidal interferences, or some other signal sets superimposed on the set of desired and known signals, we use a similar procedure.

It should be stressed that the LAM method presented here does not pretend to be a general image restoration method applicable to an arbitrary signal or image. The fact that the training set consists of a limited number of *a priori* known patterns suggests that it is applicable to a limited number of reconstructable signals or images. In the 2-D applications, for example, it is necessary to define a finite set of reference image pairs (\mathbf{f}, \mathbf{g}) that spans the space of all desirable images. To ensure that an arbitrary measured input signal, a signal that does not belong to the training set, is rejected by the LAM filter, the following procedure can be used. An auxiliary LAM matrix M_1 , an inverse of M , is formed with the input and output training sets exchanged (i.e., the Y matrix is formed from the vectors $\{\mathbf{g}_k\}$, while the X matrix is formed from the vectors $\{\mathbf{f}_k\}$). Next, the two systems M and M_1 are cascaded so that the output of the second system is a degraded version of the original object signal. To decide whether the given input belongs to the training set, the Euclidean distance D between the noisy measured input \mathbf{g} , and the output \mathbf{g}_0 signals

$$D = \sqrt{(1/N) \sum_{k=1}^N (\mathbf{g}_k - \mathbf{g})^2} \quad (22)$$

where N is the number of samples, is calculated. This distance is then compared to a noise standard deviation Δ ; if $D < \Delta$ the input measured sampled vector is accepted as part of the training set; otherwise the input measured sampled vector is rejected. Multiplying the measured signal \mathbf{g} , with M , a restored version of an accepted input measured sampled vector can be generated.

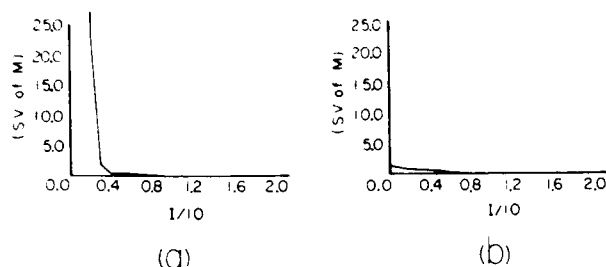


Fig. 1. (a) First twenty singular values of an unconstrained LAM matrix M ; (b) as in (a) for the constrained M ($P = 2$).

III. Experimental Results

In this section, computer simulation results for the problem of resolving two impulses as well as to reconstruct superresolving signal waveforms from a measured noisy output of ILPF are presented. The measured, with different values for their SBP, sampled vectors have various levels of uniform additive noise.

First, the results for the superresolution of two-point objects simulated by two ideal impulses are presented. To obtain the training set, a set of two-point object sampled vectors $\{f_k\}$, $k = 1, 2, \dots, 100$, so that every sequence contains two ideal impulses, is formed. The impulses are arranged so that the k th sampled vector is $f_k = (1, 0, \dots, 0, 1, 0, \dots, 0)^T$. Note as a reference that the first impulse is always at the first slot, while the second impulse shifts. The impulse strength is arbitrarily set to be unity. The data matrix Y , with its column sampled vectors f_k , is

$$Y = [f_1, f_2, \dots, f_{100}]. \quad (23)$$

To obtain the set of the training vectors $\{g_k\}$, every two-point object sampled vector f_k is passed through an ILPF. From the ILPF responses, a set of 100 training vectors $\{g_k\}$ is formed. For convenience, the number of vectors in the training set $\{g_k\}$ is taken to be equal to the number of elements of the training vectors. Using this set, the columns g_k form the key matrix X :

$$X = [g_1, g_2, \dots, g_{100}]. \quad (24)$$

The order of the columns of the matrix X corresponds to the order of the columns of the matrix Y . The input and output vectors are related by Eq. (10). As described in Sec. II [see Eq. (16)], next the constrained LAM matrix M_k is formed. The initial matrix M_0 is the noise-free covariance matrix [see Eq. (20)].

As an example, in Fig. 1(a), the first twenty singular values of the matrix M , after the first iteration step with $P = 0.2$, are presented. The rest of the singular values are nearly zero. It is clear that the condition number $c(M)$ is a large number. Figure 1(b) presents the first twenty singular values of the constrained matrix M . We note that the constraining procedure does not considerably affect the smallest singular value $\delta_{\lambda}(M)$ (which is of the order of 0.003 for this particular case). Comparing Fig. 1(a) with Fig. 1(b), since $c(M)$ has decreased, the dynamic range of the constrained LAM matrix M elements must also decrease. Figure 2

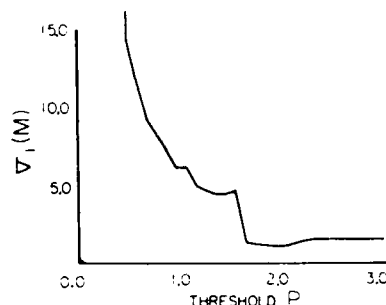


Fig. 2. Quadratic matrix norm of a superresolving matrix M as a function of the threshold parameter P .

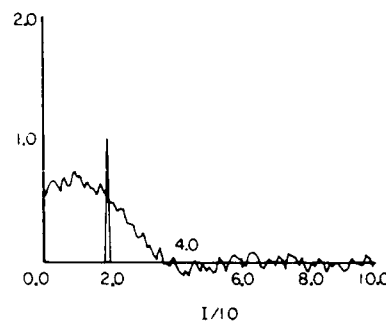


Fig. 3. Resolution of a degraded noisy two-point object with a SNR of 19 dB and SBP of 1 obtained with constrained M with singular values as in Fig. 1(b). This constrained matrix M is used in the next examples for two-point object superresolution.

presents the plot of $\|M\|$ as a function of the threshold parameter P . It is clear that, for this particular key matrix structure, the quadratic matrix norm $\|M\|$ [and consequently the condition number $c(M)$] shows a minimum for P of ~ 2 . The search for the optimum threshold parameter P has been initiated at $P_0 = 0.2$ with the increments $\Delta P = 0.2$. To ensure that a minimum has been found, because the quadratic norm of M is mostly a monotonically decreasing function of P (see Fig. 2), it was sufficient to proceed with only two iteration steps past the minimum. Figure 3 presents the result of a signal restoration using the constrained LAM matrix M whose singular values are shown in Fig. 1(b).

In Figs. 4(a) and (b), the results of superresolving two impulses from noisy measured sampled vectors, with SBP of 0.15 and two different levels of additive noise, SNRs of 19 and 13 dB, respectively, are presented. While there is some discrepancy in the strength of the impulses, they clearly illustrate the noise resistant behavior of the LAM inverse filter. In Fig. 4(c), the result of superresolving restoration of two impulses with a SBP of 0.3 and SNR of 7 dB is presented. This result demonstrates the exceptional robustness of the LAM inverse filter.

Two-dimensional images can be reduced to a long 1-D signal by concatenating the rows of a 2-D image.³¹ As an example, a 16×16 two-point object image is formed. The two ideal impulses are distributed so

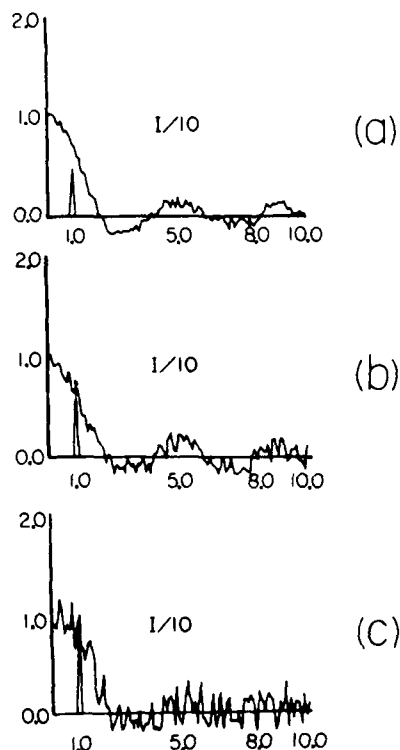


Fig. 4. (a) Superresolution of two-point objects with SBP = 0.15 and SNR of 19 dB; (b) with SNR of 13 dB; (c) with SNR of 7 dB and SBP of 0.3.

that the first impulse is kept fixed, while the second assumes all possible 256 positions. These 256 different images represent the training set of the two-point 2-D object function $\{f_k\}$. To obtain a set of degraded images $\{g_k\}$, the f_k terms are passed through a 2-D ILPF. In Fig. 5, some samples of the training degraded images $\{g_k\}$ are presented. To form the 2-D constrained LAM matrix, the previous 1-D analysis is followed. In Fig. 6, the result of superresolving restoration of a noisy degraded image with a SNR of -6 dB and SBP of 0.5 is presented. As can be observed, the result of superresolving the resolution of a 2-D image, even in the presence of high-level additive noise, is very good.

To restore more realistic superresolving images, first, a large image is decimated, say as 50×50 pixels. In this case a LAM training set vector consists of 2500 elements. The resulting LAM matrix would have at most 6.25M entries. The training set [reference pairs (f, g)] in this case may contain any number of vector pairs between 1 and 2500. In fact, in Ref. 23, a larger image set has been used.

In addition to restoring impulses, the constrained LAM matrix technique can be used to reconstruct superresolving signal waveforms. In the next examples, a set of three different types of sampled waveform, a triangle with different slope edges, a trapezoid, and a Gaussian pulse of different width (see Fig. 7), is used. The input training set $\{(f_1), (f_2), (f_3), (f_4)\}$ now

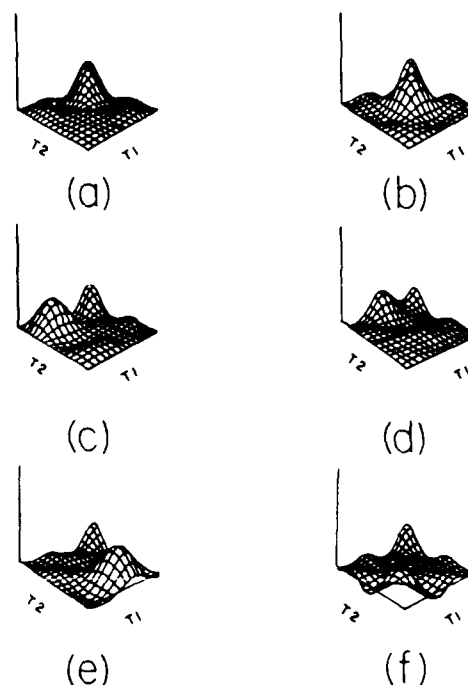


Fig. 5. Sample of the noise-free degraded training patterns used in constructing the 2-D constrained M matrix.

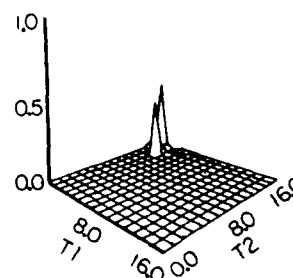


Fig. 6. Result of the restoration of a 2-D two-point object with an SNR of -6 dB and SBP of 0.5.

consists of triangles $Y_1 = \{f_1\}$, trapezoids $Y_2 = \{f_2\}$, Gaussian pulses $Y_3 = \{f_3\}$, with different slope edges and different width, and a set of zero vectors $Y_4 = \{f_4\}$, with thirty-two sampled vectors in each subset. Corresponding to the set of zero vectors $\{f_n\}$, the $\{g_n\}$ is a set of pure noise vectors. Noise sampled vectors $\{g_n\}$ and their counterparts, the zero vectors $\{f_n\}$, are introduced into the training set to provide the noise rejection property for the LAM matrix by forcing production of a null as its output response to noise. Each data sampled vector consists of 128 elements. This set is used to form the data matrix Y :

$$Y = [Y_1, Y_2, Y_3, Y_4]. \quad (25)$$

The set of output training vectors $\{(g_1), (g_2), (g_3), (g_4)\}$ ($X_1 = \{g_1\}$, $X_2 = \{g_2\}$, $X_3 = \{g_3\}$, $X_4 = \{g_4\}$) is obtained from the response of ILPF to the input training set

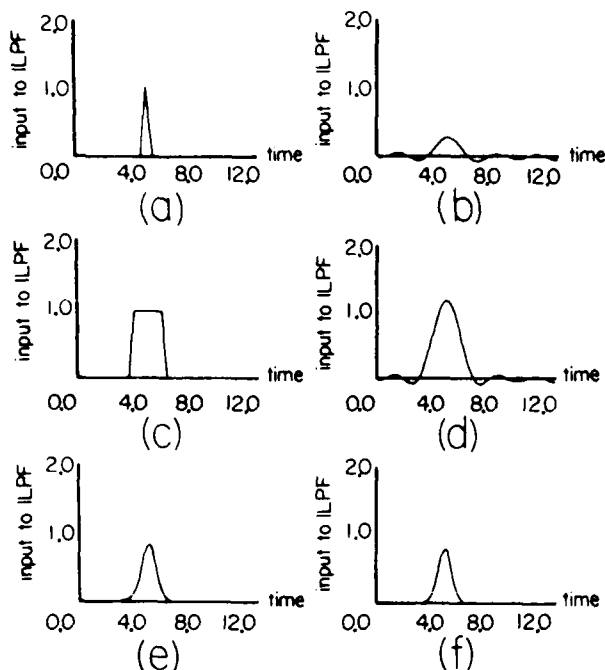


Fig. 7. Sample of the input-output noise-free training patterns (input-output of ILPF) used to form a LAM matrix M : (a) and (b), a triangle; (c) and (d), a trapezoid; (e) and (f), a Gaussian pulse.

$\{(f_s), (f_t), (f_p)\}$ (see Fig. 7). The output training vectors form the key matrix X ,

$$X = [X_1, X_2, \dots, X_N] \quad (26)$$

The input and output training sampled vectors are related by the matrix Eq. (10), which in this case is a 128×128 square matrix.

The restoration of the original sampled waveform is performed multiplying a noisy degraded sampled waveform, as in Figs. 8(a), (c), and (e), by M_{128} . The results of restoration are presented in Figs 8(b), (d), and (f). In the presented examples, the restoration has been performed in the presence of the same level of additive noise and different SBP for every particular type of waveform. In Fig. 8(b) the result of superresolving restoration of a degraded noisy triangle with SBP of 0.03 and SNR of -3 dB is presented. Figure 8(b) shows that for heavy degradation (SBP of 0.03) and SNR of -3 dB the result of restoration is very good. In Fig. 8(d) the result of restoration of a noisy degraded trapezoid for SBP of 0.15 and a SNR of 9 dB is presented. It can be observed that the edges and corners of the trapezoid are well reconstructed. Finally, in Fig. 8(f) the results of superresolving restoration of a Gaussian pulse for a SBP of 0.08 and SNR of 2 dB are presented.

IV. Summary and Conclusion

A new approach to superresolving signal and image restoration has been proposed. At the expense of limiting the set of reconstructable signals, an excep-

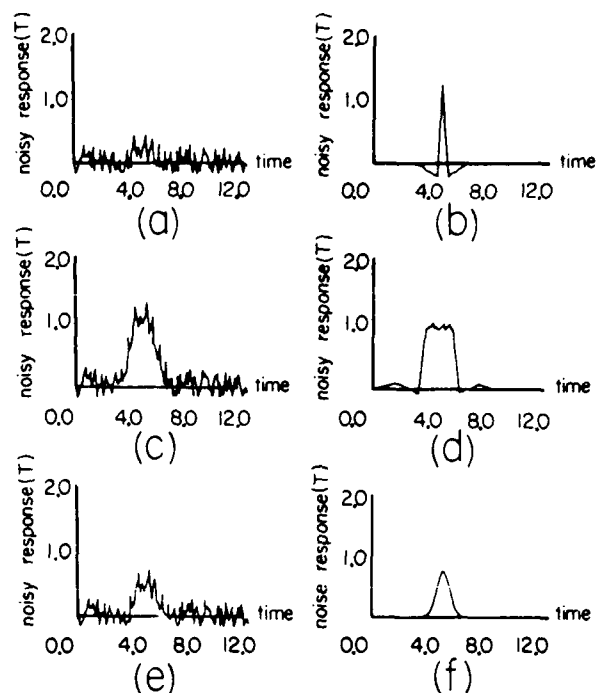


Fig. 8. (a) Noisy degraded triangle from Fig. 7(a); SNR = -3 dB, SBP = 0.03; (b) the result of restoration of the signal in (a); (c) noisy degraded trapezoid from Fig. 7(c), SNR = 9 dB, SBP = 0.15; (d) the result of restoration of the signal in (c); (e) the noisy degraded Gaussian pulse from Fig. 7(e), SNR = 2 dB, SBP = 0.08; (f) the result of restoration of the signal in (e).

tionally robust inverse filter has been obtained. The filter is based on the principle of the content addressable memory (CAM). A practical realization of the CAM was performed by employing a linear associative memory technique. To determine the LAM matrix operator, a constrained training algorithm has been introduced. The inverse filter was tested on different sets of 1-D and 2-D degraded signals with different levels of degradation and in the presence of very significant noise. The 1- and 2-D noisy degraded two-point sources have been resolved for SBP far below unity. It has been shown that the proposed inverse filter restores equally well both impulsive and edge-type images. The presented method limits the set of applicable signals to a known training set. However, the set of desirable signals can be extended by introducing several LAM parallel inverse filters trained to a particular set of signals. A mechanism to reject statistically undesirable signals, as well as the signals that are not in the training set, has been introduced. Once calculated, for a given set of training signals, the entries of the LAM matrix M could be stored in either a set of high-density read-only memory chips or an optical holographic memory for further use. With a fast optical and/or electronics vector processor (it is necessary to perform only one matrix-vector multiplication), the LAM inverse filter is easily hardware realizable. This suggests its use in a real-time environment.

This work was supported in part by a grant from Air Force Office of Scientific Research. Some of this material was presented at the 1985 OSA Annual Meeting, Washington, DC.

References

1. R. Barakat and G. Newsam, "Algorithms for Reconstruction of Partially Known, Band-Limited Fourier-Transform Pairs from Noisy Data," *J. Opt. Soc. Am. A* **2**, 2027 (1985).
2. R. Gerschberg and W. Saxton, "A Practical Algorithm for Determination of Phase from Image and Diffraction Plane Pictures," *Optic* **35**, 237 (1972).
3. A. Papoulis, "A New Algorithm in Spectral Analysis and Band-limited Extrapolation," *IEEE Trans. Circuits Syst.* **CAS-22**, 735 (1975).
4. D. C. Youla, "Generalized Image Restoration by the Method of Alternating Orthogonal Projections," *IEEE Trans. Circuits Syst.* **CAS-25**, 695 (1978).
5. J. A. Cadzow, "An Extrapolation Procedure for Bandlimited Signals," *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-27**, 4 (1979).
6. M. S. Sabri and W. Steenart, "An Approach to Bandlimited Extrapolation: The Extrapolation Matrix," *IEEE Trans. Circuits Syst.* **CAS-25**, 74 (1978).
7. R. W. Schafer, R. M. Mersereau, and M. A. Richards, "Constraint Iterative Restoration Algorithms," *Proc. IEEE* **69**, 432 (1981).
8. B. R. Frieden, "Image Enhancement and Restoration," in *Picture Processing and Digital Filtering*, T. S. Huang, Ed. (Springer-Verlag, New York, 1975).
9. P. A. Jansson, Ed., *Deconvolution with Applications in Spectroscopy* (Academic, Orlando, FL, 1984).
10. C. W. Helstrom, "Image Restoration by the Method of Least Squares," *J. Opt. Soc. Am.* **57**, 297 (1967).
11. A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems* (Winston and Sons, Washington, DC, 1977).
12. B. R. Frieden, "Maximum-Information Data Processing: Application to Optical Signals," *J. Opt. Soc. Am.* **71**, 294 (1981).
13. S. J. Howard, "Method for Continuing Fourier Spectra Given by the Fast Fourier Transform," *J. Opt. Soc. Am.* **71**, 95 (1981).
14. S. J. Howard, "Continuation of Discrete Fourier Spectra Using a Minimum-Negativity Constraint," *J. Opt. Soc. Am.* **71**, 819 (1981).
15. R. J. Mammone and G. Eichmann, "Superresolving Image Restoration Using Linear Programming," *Appl. Opt.* **21**, 496 (1982).
16. C. K. Rushforth and R. L. Frost, "Comparison of Some Algorithms for Reconstructing Space-Limited Images," *J. Opt. Soc. Am.* **70**, 1539 (1980).
17. C. K. Rushforth, A. E. Crawford, and Y. Zhou, "Least-Squares Reconstruction of Objects with Missing High-Frequency Components," *J. Opt. Soc. Am.* **72**, 204 (1982).
18. H. Stark, D. Cahana, and H. Webb, "Restoration of Arbitrary Finite-Energy Optical Objects from Limited Spatial and Spectral Information," *J. Opt. Soc. Am.* **71**, 635 (1981).
19. J. Meade and K. Murata, "Restoration of Band-limited Images by an Iterative Regularized Pseudoinverse Method," *J. Opt. Soc. Am. A* **1**, 28 (1984).
20. D. Slepian and H. O. Pollak, "Prolate Spheroidal Wave Functions, Fourier Analysis and Uncertainty I," *Bell Syst. Tech. J.* **40**, 43 (1961).
21. A. Papoulis, *Signal Analysis* (McGraw-Hill, New York, 1977).
22. C. T. Baker, L. Fox, U. F. Muiers, and K. Wright, "Numerical Solution of Fredholm Integral Equations of the First Kind," *Comput. J.* **7**, 141 (1964).
23. T. Kohonen, *Self-Organization and Associative Memory* (Springer-Verlag, Berlin, 1984).
24. A. Albert, *Regression and the Moore-Penrose Pseudoinverse* (Academic, New York, 1972).
25. G. S. Stiles and Dong-Lih Denq, "On the Effect of Noise on the Moore-Penrose Generalized Inverse Associative Memory," *IEEE Trans. Pattern. Anal. Machine Intell.* **PAMI-3**, 358 (1985).
26. G. H. Golub and C. F. Van Loan, *Matrix Computations* (Johns Hopkins Press, Baltimore, 1983).
27. H. C. Andrews and C. L. Patterson, "Singular Value Decomposition and Digital Image Processing," *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-24**, 26 (1976).
28. T. N. E. Greville, "Some Applications of the Pseudoinverse of a Matrix," *SIAM Rev.* **2**, 15 (1960).
29. L. D. Pyle, "Generalized Inverse Computation Using the Gradient Projection Method," *J. Assoc. Comput. Mach.* **11**, 422 (1964).
30. B. Rust, W. R. Burrus, and C. Schneeberger, "A Simple Algorithm for Computing the Generalized Inverse of a Matrix," *Commun. ACM* **9**, 381 (1966).
31. H. C. Andrews and B. R. Hunt, *Digital Image Restoration* (Prentice-Hall, Englewood Cliffs, NJ, 1977).

Topologically Invariant Texture Descriptors

G. EICHMANN AND T. KASPARIS

*Department of Electrical Engineering, City College of the City University of New York,
New York, New York 10031*

Received August 4, 1986; accepted July 10, 1987

Texture is one of the important image characteristics and is used to identify objects or regions of interest. The problem of texture classification has been widely studied. Texture classification techniques are either statistical or structural. Some statistical texture classification approaches use Fourier power-spectrum features, while others are based on first- and second-order statistics of gray level differences. Periodic textures that consist of mostly straight lines are of particular interest. In this paper, a new structural approach based on the Hough method of line detection is introduced. This classification is based on the relative orientation and location of the lines within the texture. With proper normalization, the classification is independent of geometrical transformations such as rotation, translation, and/or scaling. Experimental results will also be presented. © 1988 Academic Press, Inc.

1 INTRODUCTION

Texture is one of the most basic properties of a visible surface. The visible image texture may be due to changes on the surface of an object, may be segments of lines as on a brick wall, or simply could be due to a collection of objects as in an aerial photo. Texture features are useful for both image analysis and classification, as well as in scene segmentation and identification, such as in an image understanding system applied to robot vision, industrial inspection, and photo analysis [1, 2].

In this paper, a new structural line detection approach is presented. The rationale for this approach is that regular textures consist of an arrangement primarily of line structures appearing periodically in the texture. Furthermore, it has been postulated that a human eye classifies structures based on line detection; i.e., the human eye is primarily a line detector [3, 4]. In our approach, line detection is accomplished using the Hough technique [24]. The Hough transform (HT) maps line segments into a point in the transform domain. In the HT domain, the line length can also be obtained. The method consists of calculating the texture HT and extracting features used for classification. Texture features that can be obtained from the Hough domain are principal directions of lines in the texture, periodicity and line separation in each direction, etc. With proper normalization, classification is not affected by geometrical transformations such as rotation, translation, or scaling.

The paper is organized as follows: In Section II the background on the previous work will be presented and then, in Section III, the Hough transform (HT) will be introduced. Some useful HT properties will be discussed, and, in addition, an efficient HT computation algorithm will be given. In Section IV, the topologically invariant texture classification problems will be discussed and the classification algorithm will be presented. In Section V, experimental results and examples will follow, with some concluding remarks.

II. BACKGROUND

Texture may be considered either as a pattern of different spatial intensity arrangements, or as a basic periodically or quasi-periodically repeated local pattern. This definition is applicable to line patterns, such as ruled-line arrays and tiling patterns. Texture also relates to the spatial size of the tonal primitives of an image. A larger size tonal primitive shows a coarse texture, while a smaller size tonal primitive indicates a fine texture. The texture autocorrelation function, basically its spatial frequency content, determines the size of tonal primitives. An alternative approach is to view texture not in terms of spatial frequency content but in its edgeness per unit area [5]. Coarse textures have a small number while fine textures have a large number of edgeness per unit area. Rosenfeld and Thurston [6] used the average value of the texture gradient as a local image property. Sutton and Hall [7] extended this idea by considering the gradient to be a function of the distance between the pixels. This last gradient method is directly related to the texture autocorrelation function.

The problem of texture classification has been studied for many years and a number of approaches have been developed. Analysis and classification are based on texture features which are derived using either statistical or structural methods, or both in a combined manner. Statistical approaches view texture regions as a sample of a two-dimensional stochastic process. This process is describable by its statistical parameters. This model-based formulation is well suited for natural textures consisting of segments of multigray level images such as grass, sand, and wool. Structural approaches are based on the view that textures are made up of primitives which appear in nearly regular repetitive spatial arrangements. To structurally specify a texture, the primitives and the placement rules [8] must be described.

Some of the statistical approaches include autocorrelation functions [5], textural edgeness [6], spatial gray level co-occurrence probabilities [9], and gray level run lengths [10]. The Fourier power spectrum of a texture gives essentially its statistical information [11], although it has been used to determine some structural descriptions, such as its periodicity and its directionality [12].

Structural approaches are based on the view that within the texture there is a regular repetition cell. To describe texture, one needs to prescribe the resolution cell and the various placement rules with which the texture is formed. This approach is especially suited for regular, periodic textures. Zucker [9, 13] suggested that natural textures be viewed as a distortion of an ideal regular structure. Carlucci [14] described a texture model using line segments and/or polygons as primitives in which the placement rules are given syntactically in a graph-like language. Lu and Fu [15] presented a tree grammar syntactic approach for texture. The basic difference, in various structural approaches, is the choice of primitives. These can be either pixels [15], gray level peaks [16], line segments [14], or tiles [17]. Statistical methods, with their primitives such as edgeness per unit area or run lengths, can be combined with structural approach. In other structural approaches, the primitives used are the average edge separation in different orientations [18] and the repetition of edges in different orientations, by the calculation of edge repetition arrays [19].

A survey of texture analysis methods can be found in [8, 20, 21]. Several examples of natural textures can be found in Brodatz's photographic album of natural textures [22].

III. THE HOUGH TRANSFORM

The Hough transform [23] (HT) is a method for detecting curves that can be described by a number of parameters, such as lines [24], circles [25], and parabolas [26, 27]. The HT is used to compute the locus, in parameter space, of the set of curves passing through a point in the image plane. In the special case where line detection is desired, since two parameters are sufficient to specify a straight line, the parameter space is two-dimensional. Hough [23] chose to use the slope and the intercept as the two parameters. Because these parameters are unbounded, the application of the technique is complicated. Through the so-called normal parametrization, a bounded parameter set is obtained. As illustrated in Fig. 1, a straight line (L) is specified by the angle formed by the x -axis and the normal to L , and its distance p measured from the origin to the point where the normal and L intercept. The equation of the line L is

$$p = x \cos \theta + y \sin \theta. \quad (1)$$

Restricting θ to the interval $[0, \pi]$ leads to unique normal parameters $[p, \theta]$, i.e., every line in the $[x, y]$ plane corresponds to a unique point in the $[p, \theta]$ plane. From Eq. (1), we note that a point in the $[x, y]$ plane corresponds to a sinusoidal curve in the $[p, \theta]$ plane. Suppose now that we have a set of points $[x_i, y_i]$ lying on a straight line. For every one of these points there corresponds a sinusoidal curve in the $[p, \theta]$ plane, specified by

$$p = x_i \cos \theta + y_i \sin \theta.$$

We can show that all these curves corresponding to points of a straight line have a common point of intersection, say (p_0, θ_0) in the transform plane. This point defines the line that passes through all these points. In summary, we have the following HT properties:

PROPERTY 1. A point in the image plane corresponds to a sinusoidal curve in the $[p, \theta]$ transform plane.

PROPERTY 2. A point in the $[p, \theta]$ transform plane corresponds to a straight line in the image plane.

PROPERTY 3. Points on the same curve in the $[p, \theta]$ transform plane correspond to lines through the same point in the image plane.

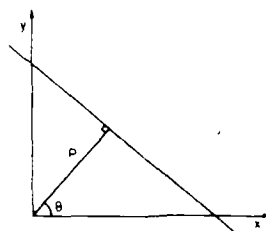


FIG. 1. The normal parameters of a straight line.

PROPERTY 4. Points on the same line in the image plane correspond to curves through the same point in the $[p, \theta]$ transform plane.

We next apply these results to line detection. First, both p and θ are quantized to N_p and N_θ levels, respectively, with levels that depend on the desired resolution. The required computation increases linearly with N_θ or with an increasing number of image points, but it does not depend on N_p . A larger N_p increases the storage requirements, but not the amount of computation. For N_θ levels in θ and N_p levels in p , there will be $N_\theta \times N_p$ quadrupled grid points in the $[p, \theta]$ plane. In the case of binary images, we assume that a zero corresponds to dark background while unity corresponds to bright points in the image. The quantized $[p, \theta]$ region is treated as a two-dimensional (2D) array of accumulators, initially set to zero. The transform procedure is to scan all nonzero pixels in the image. At every nonzero pixel, at position (x_i, y_i) , for every quantization value of θ , we compute p and for every pair $[p, \theta]$ we increment the corresponding accumulator. Thus, a given cell in the 2D accumulator array eventually records the total number of curves passing through it. When all image pixels are scanned, the value in each $[p_i, \theta_i]$ cell yields the number of pixels (within quantization error) which lie on the line $[p_i, \theta_i]$. Therefore, large peaks in the accumulator correspond to long lines in the image. If there are n nonzero pixels in the image, then for each pixel N_θ calculations are required and overall $n \times N_\theta$ calculations are needed to complete the transformation. Clearly when n is large, compared to an exhaustive search that requires considering the lines between all $\frac{1}{2}(n(n-1))$ pairs of image pixels, this procedure is more efficient.

There are limitations to this approach. First, the final results are sensitive to both N_θ and N_p . While a finer quantization increases resolution, it also exposes the problem of clustering entries corresponding to nearly colinear points. Furthermore, as was mentioned earlier, a finer quantization in N_θ will also increase the computation time. Second, this technique finds colinear points without regard to their contiguity. Thus, a peak value in the HT domain may represent either a continuous or a number of smaller discontinuous segments on the same line. Furthermore, an image line segment can be distorted by unrelated image pixels. Finally, in an image containing many lines of different orientations, there will be "crosstalk" among these lines in the HT plane. Consider, for example, the case of an image where there are twenty lines parallel to the y axis. Then, depending on the line-to-line separation in the $[p, \theta]$ domain, there will be twenty peak values in the $\theta = 0^\circ$ axis for different p values. However, in searching for lines parallel to the x axis, we will always find twenty colinear points. Thus, in the $[p, \theta]$ domain, we will find smaller peaks in the direction of $\theta = 90^\circ$, and for all values of p . This problem is even more exaggerated in the case of a natural scene where the lines are not just one but many pixels wide.

In the previous discussion, we have considered binary, i.e., only two, gray level images. A natural scene is usually digitized to more than two gray levels and modification is necessary in order to adopt the Hough technique. There are two possible ways to accomplish it. The first method is to threshold the image so that only two gray levels will exist. However, information is lost during thresholding. In the case of a regular texture, after thresholding, while some detail is lost, the basic texture structure is still present. The second method is to relate the HT to the Radon transform (RT) [28, 29]. The RT is a well-known integral transform from the theory

of computed tomography. The forward RT is defined as

$$F(p, \theta) = \int \int_{\mathbb{R}^2} f(x, y) \delta(p - x \cos \theta - y \sin \theta) dy dx, \quad (2)$$

where $f(x, y)$ is the input, $F(p, \theta)$ is the transform and $\delta(\cdot)$ is the unit Dirac impulse function. $F(p, \theta)$ is equivalent to the integration of $f(x, y)$ along a straight line whose normal parameters are $[p, \theta]$. Recently, it has been pointed out that, for binary images, the forward RT is equivalent to the HT [30]. We extend this concept by applying the RT to the nonbinary image and by approximating the integral of the image along a line. The simplest approximation is to add along a given line all the pixel intensities. Thus, when we calculate the transform by updating the 2D register and when we get to the point to update a cell, we do not just increment by one, but we add to it the intensity of the pixel under consideration. In this sense, brighter lines correspond to brighter peaks in the RT domain.

More advanced HT implementations are found in [31]. Optical implementation of the HT, both coherent and incoherent versions, is also possible [32, 33]. Recently, an optical HT that works at real-time video rates has been reported [34].

IV. TEXTURE CLASSIFICATION

Structural approaches to texture classification are based on regularity. A resolution cell is periodically repeated within the texture in accordance with some placement rules. This approach is very well suited for regular, periodic structures. A different point of view of periodic structures is to consider them as a regular arrangement of lines, or line segments, of different orientations. Consider, for example, an image of a brick wall. Such a structure can be viewed as a repetition of horizontal lines, with line segments regularly arranged in the vertical direction. However, if a rotated view of the same image is considered again, then horizontal and vertical lines might no longer exist. In this case, we can still view this structure as a regular repetition of solid lines of some orientation, with line segments perpendicular to the solid lines. This last view is independent of rotation, a very desirable property of a classifier.

In some structural approaches, the primitives used for classification are the average separation of edges at some certain orientations and the arrangement of edges, again in certain orientations [18, 19]. However, the number of orientations to be considered is limited by the required computation, and the classification is rotation sensitive. A model based, rotation invariant texture classification method is described in [35].

The classification method proposed in this paper does not consider line (or edge) orientation, but instead the angle at which they intersect. This angular feature does not depend on image rotation. A second classification primitive is the normalized separation of lines for the same orientation. This feature is independent of scale, since the normalized separation does not change with scale. Furthermore, this feature is also independent of linear image translation. A third classification primitive is the number of principal line orientations within a texture. In a regular texture, the lines or edges will not appear along random directions, but will be arranged regularly at some orientations. For example, in a brick wall image, the lines appear along two principal directions spaced 90° apart.

These primitives are especially suited to describe regular textures that consist mainly of straight line elements. Lines, or more generally curves, are an important characteristic of textures. The HT is a very efficient technique to find desirable texture primitives, since all the information can be extracted from the $[p, \theta]$ domain. Since the classification algorithm needs to be invariant under geometrical transformation, such as rotation, translation, or scaling, therefore how the HT of an image is affected by these transformations needs to be examined. In this regard the following HT properties are of interest.

Assuming that an infinite periodic line structure exists, then the following HT properties hold:

PROPERTY 1. If two lines intersect at angle ϕ , where $0^\circ < \phi \leq 90^\circ$, then the corresponding points in the $[p, \theta]$ domain will be also located ϕ° apart.

PROPERTY 2. Linear translation of the image has the effect of shifting the HT in the p direction. The shift of the transform is not uniform, i.e., some peaks at some angles may shift less, or even not shift at all. This is evident from the fact that the normal parameters of lines parallel to the direction of the translation will not change, whereas the parameters of the lines perpendicular to the direction of the translation will be most affected.

PROPERTY 3. Linear translation does not affect the spacing between peaks at a given angle. This is so because, after the translation, while the location of lines with respect to the origin may change, the relative line-to-line spacing does not change.

PROPERTY 4. Image rotation, by some angle ϕ , is equivalent to a circular shift of the HT by the same angle ϕ . This property is due to the fact that lines that are located at angle θ_0 , after rotation will be located at angle $\theta_0 + \phi$. The shift is circular because all points crossing the 180° axis will again appear at the 0° axis, i.e., the HT is "wrapped" around a cylinder, so that the $\theta = 0^\circ$ axis is identical to $\theta = 180^\circ$ axis.

PROPERTY 5. Scaling the texture in all directions (zooming) is equivalent to scaling the HT in the p -direction. This is so because, after the scaling, for all lines the p -parameters will scale, but not the θ -parameters.

These HT properties show the effects of geometrical transformations on the HT and also indicate that with proper normalization it is possible to compensate for all geometrical image transformations.

Zucker [13] suggested that natural textures can be considered as distortions of ideal structures. What would appear to be ideal are structures formed of thin straight lines, periodically arranged in different directions. It was pointed out earlier that there is some "crosstalk" between lines at different directions. This is more pronounced in textures where there are wide lines present. In this case, the HT peaks corresponding to these lines will not be single points but rather regional peaks. Also, all these curves will have the effect of raising the values of other accumulators in the 2D register, and therefore the HT will have the appearance of wide peaks in a noisy background. However, in the ideal case, where the image lines are one pixel wide, then the curves corresponding to line pixels will pass through

one point in the HT domain. In this case, the HT will have the appearance of large sharp peaks in some low-level noisy background. It is apparent that line detection now is far more accurate. From this discussion, it follows that classification accuracy can be increased by preprocessing the natural texture so that it will be as close as possible to an ideal one, or in other words, by extracting the "skeleton" of the texture. This can be accomplished by thinning the lines or by first applying an edge detector and then applying a gap filling operator to form solid lines. Different techniques are available for this purpose [21, 36].

In the following discussion, it will be assumed that the texture is a regular, periodic structure formed by thin straight lines. This image can be the result of preprocessing, or simply it is an ideal test pattern. The next classification step is to calculate the HT to a desired resolution. Texture HT primitives will then be extracted. The HT texture primitives are: (a) the number of line orientations, (b) the relative orientation angles, and (c) the line spacing for each orientation. Several normalization procedures will be employed so that: (a) classification will not depend on geometrical transformations and (b) the dimensionality of the feature space is reduced. Furthermore, to reduce the space to 3D, some statistical parameters will also be used.

The classification algorithm is composed of the following steps: First calculate the texture HT with the desired resolution. As mentioned earlier, finer quantization of the p -axis does not increase the computation, but only the storage requirements. Therefore the important decision is to select the quantization levels of the θ -axis, i.e., the directions at which we will search for lines. Next, locate and isolate peak values in the HT. Different approaches can be used at this step. One approach is to simply threshold the HT with a proper threshold parameter so that only the peak values remain. However, simple thresholding exposes the problem that peaks below the threshold will not be "seen," whereas in some regions of the transform the background noise may be larger than the peaks in a different region. Therefore the proper threshold value is very critical, and must be very carefully selected. A better approach is to use a median filter (MF) [37]. The MF will eliminate local peaks in the HT. Hence, subtracting the output of an MF from the original transform will reveal the local peaks. The window of the MF should be kept sufficiently small so that the peaks could be eliminated without much affecting the background. Some post thresholding will eliminate any residual noise. Since the HT peaks are usually regional, it is very possible that neighboring angles will also have peak value. For example, if a texture has lines oriented at $\theta = 30^\circ$ it is possible that peak values will be also found at, say, $\theta = 28^\circ$ and $\theta = 32^\circ$. Since these angles actually correspond to the same lines, these peak values should be merged with the central peaks. Then, for every value where peaks are present, record that value and the distance between consecutive peaks. The result of this step is a table listing those angles where peaks were found and the distance from peak to peak. This is the information about line orientation and separation of lines at each direction. However, this information depends on geometrical transformations and it must be normalized. Therefore, normalize all the recorded distances in the table to the maximum or the minimum one. Usually the maximum is the better choice, since it is less sensitive to distance errors. This normalization will take into account texture scaling. Next, at every angle calculate the average of the normalized distances. The average normalized separation at every orientation was found to be a good measure of the arrangement

of the lines at that orientation. After this step all the information is in a table of two rows and as many columns as there are orientations in the texture. The first row is the value of angles where peaks were found, and the second row is the average normalized separation at those angles. Now, circularly shift the columns of this table so that the maximum value of the second row occurs at the first column. This way we select the orientation with the largest average separation as the reference orientation. Then, subtract the first entry of the first row from all the entries at the first row. Subtraction is modulo-180°. This will set the reference orientation at $\theta = 0^\circ$ and take rotation into account. Next, divide all entries of the first row by 180°. This will normalize the angles to 180°. Also, divide all entries in the second row by the first entry of the same row. This will set the maximum average separation to unity. After the previous steps, all texture information is condensed into a normalized table of two rows and as many columns as there are texture orientations. Note that the first column of the normalized table contains no information since it has the values of zero and one. We can consider the two rows of the table as two feature vectors, one revealing information about line directionality and one about line separation. For multidirectional textures, the dimension of these feature vectors will be large. We can reduce the dimension of the feature space by replacing the two vectors by their corresponding variance. This step reduces the accuracy of classification. Hence, calculate the variance of each row of the table at the previous step and form a feature vector $\mathbf{x} = [x_1, x_2, x_3]$, where

x_1 = number of orientations in the texture.

x_2 = variance of the normalized separations.

x_3 = variance of the normalized orientations.

This last feature vector will be used for classification.

The classification process consists of finding the feature vector of the texture to be classified using the previous algorithm. Then a similarity measure, or any clustering algorithm, can be used to classify the unknown vector to one of the known classes established during the learning process. Note the importance of the x_1 element of the feature vector. Textures with different numbers of orientations cannot be classified incorrectly in the same class.

V. EXPERIMENTAL RESULTS

In this section we present and discuss computer simulation results. The first step in these experiments was the computation of the Hough transform of an input texture. The HT p -axis is quantized uniformly to 361 levels while the θ -axis is quantized uniformly to 90 levels. Since the 2D register has 32,490 accumulators, it would be impractical to print the value of each accumulator, and instead, visual display on the screen of a monitor was chosen. For this purpose, the entire HT is inserted in an NTSC frame with the proper display format. The values of every row of the 2D register become pixel values of one horizontal scan line. Interlaced scanning is employed. Since the HT will not fill up the entire frame, the boundaries of the transform and the p -axis are drawn. Furthermore, divisions on the θ -axis are displayed as bright ticks. The texture to be transformed is entered into the computer through a video camera and digitized in the form of a 512×512 array quantized to

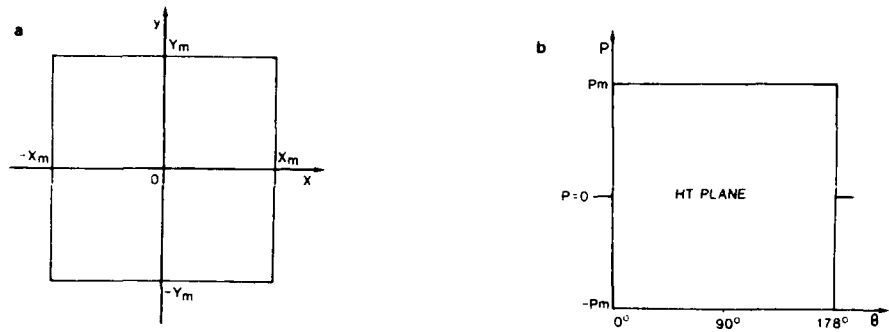


FIG. 2. (a) X-Y Plane with reference axes; (b) HT Plane with reference axes.

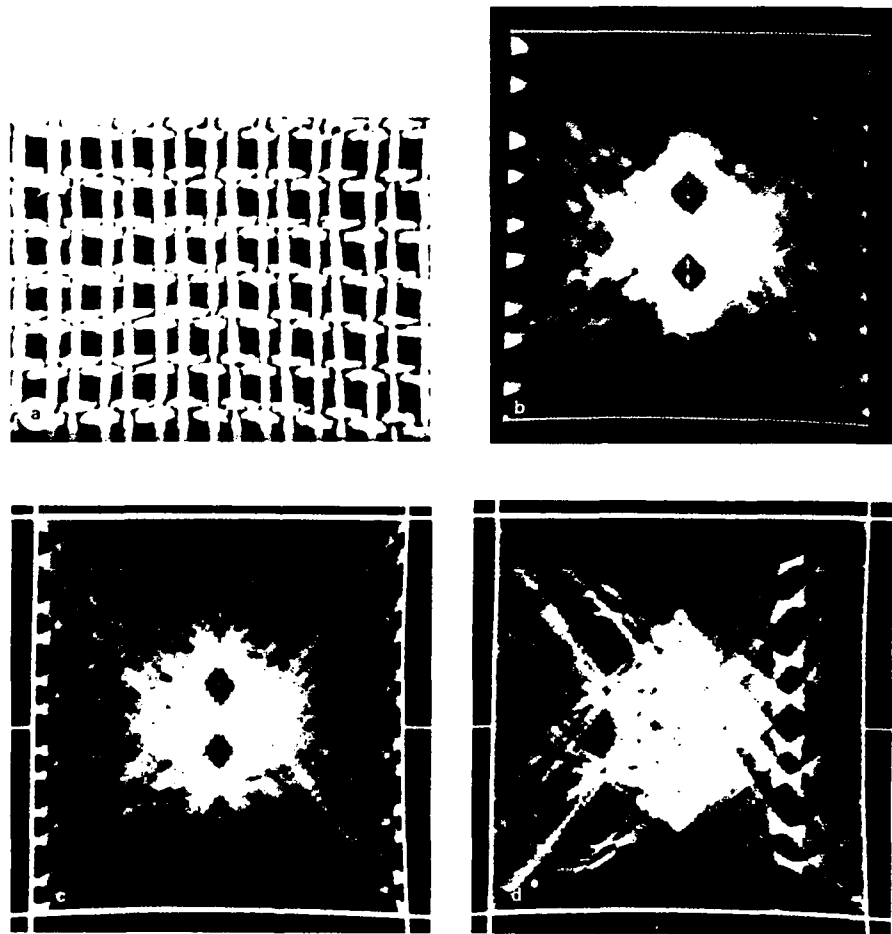


FIG. 3. (a) Texture of French canvas; (b) HT of a block of Fig. 3a; (c) HT of a block of Fig. 3a with preprocessing for line thinning; (d) HT of a rotated and scaled block of Fig. 3a with preprocessing for line thinning.

8 bits per pixel. The NTSC sync information is also included in the array. The computer used for these experiments was a DEC VAX 11/750 system.

Figure 2 shows the reference axes used in generating the HT. Figure 2a represents the texture in the x - y plane. The texture is assumed to be a rectangle of dimensions $2Xm \times 2Ym$, with the origin located at its center. By locating the origin at the center of the texture, a better utilization of the p - θ plane is possible. Figure 2b represents the p - θ (HT) plane as it will appear in the following illustrations.

Figure 3a shows the texture image of French canvas (Brodatz's plate No. 20). Notice the line structure of this texture. Figure 3b is the HT of a scaled block of this texture. It is evident that a smaller block of the texture contains the same structural information as a larger one.

In Fig. 3b the leftmost part of the transform corresponds to $\theta = 0^\circ$ and the rightmost to $\theta = 178^\circ$. The $p = 0$ axis is indicated by the horizontal solid line around the center of the transform. The bright spots around $\theta = 0^\circ$ and $\theta = 90^\circ$ correspond to peak values in the transform. Note that the spacing of the peaks follows the arrangement of the loines in the texture. Note also the background noise, especially at the center of the transform, because of "crosstalk" between lines.

As was mentioned earlier, preprocessing the texture by thinning the lines (skeletonizing), will increase the classification accuracy. Figure 3c is the HT of the same texture block used for Fig. 3b, but having undergone a process of line thinning. Comparing Fig. 3b with 3c, we see that the peak values corresponding to texture lines are located at the same position, but in Fig. 3c they are much sharper, and the background noise (crosstalk) is significantly lower. It is obvious that even though both Figs. 3b and c reveal the same information about the structure of the texture, this information can be more easily and accurately extracted from Fig. 3c. Finally, Fig. 3d is another block of the same texture under a different scale, which has been preprocessed for line thinning, and rotated by 45° . Here we note that the sharp peaks are now located around $\theta = 45^\circ$ and $\theta = 135^\circ$, where the spacing of the peaks again follow the spacing of the lines in the texture.

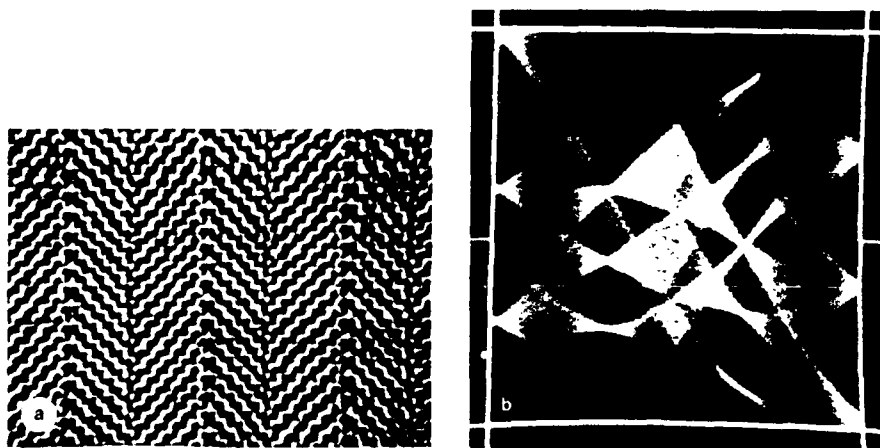


FIG. 4. (a) Texture of Herringbone weave; (b) HT of a preprocessed block of Fig. 4a.

Figure 4a is another texture of Herringbone weave (Brodatz's plate No. 17), where Fig. 4b is the HT of a preprocessed block of this texture. Notice again that the locations of the peaks correspond to the locations of the lines in the texture. We also note that since the lines around $\theta = 45^\circ$ and $\theta = 135^\circ$ are actually noncontiguous line segments which are equivalent to a shorter line, that those peaks appear to be smaller compared to the peaks at $\theta = 0^\circ$ which correspond to a long line.

It is interesting to examine what would be the effect of noise on the HT. Figure 5a is an idealized test pattern and Fig. 5b is the HT of Fig. 5a. Figure 5c shows the texture of Fig. 5a with computer generated pseudo-random uniform "salt and pepper" noise added. We can predict that the additional noise pixels will add more sinusoidal curves in the transform domain, and the result will be that more background noise will be present in the transform. Figure 5d shows the HT of the noisy texture of Fig. 5a. Comparing Fig. 5d with Fig. 5b, i.e., the HT of the texture

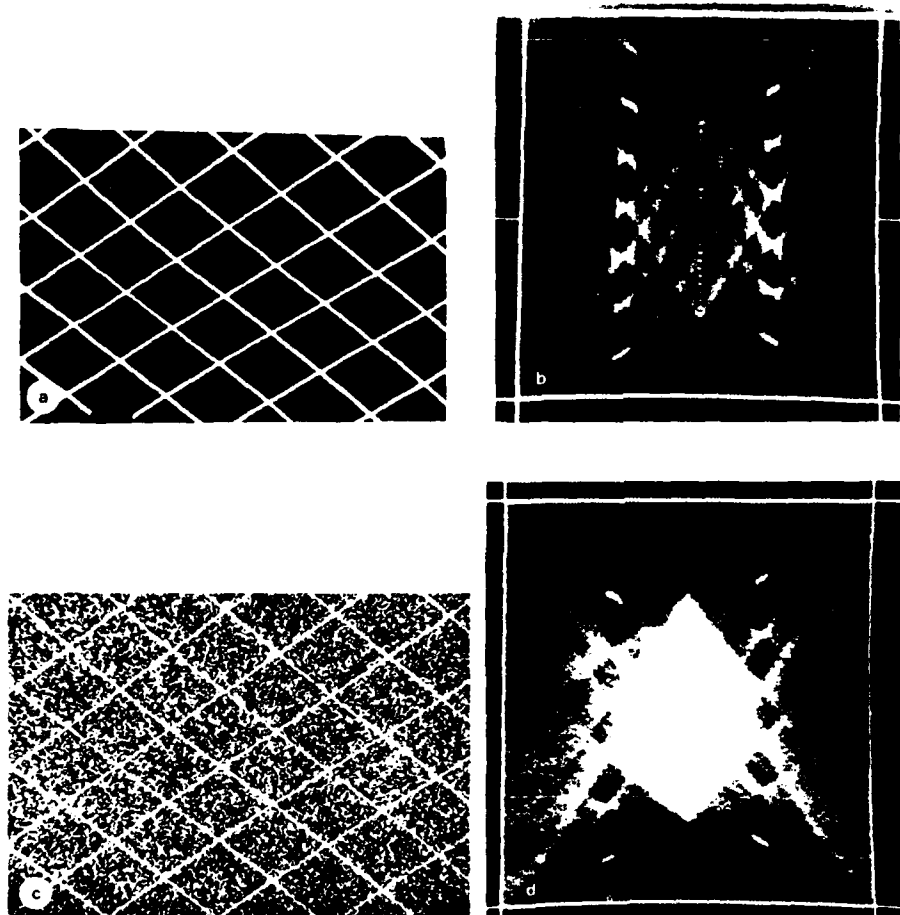


FIG. 5. (a) Test texture pattern; (b) HT of Fig. 5a; (c) Fig. 5a with uniform "salt and pepper" noise added; (d) HT of Fig. 5c.

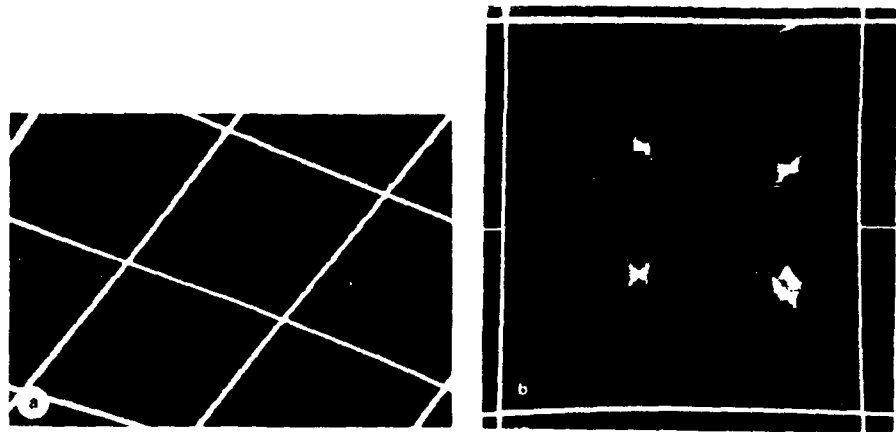


FIG. 6. (a) Fig. 5a under scaling and translation; (b) HT of Fig. 6a.

without noise, we notice that indeed the noisy transform has larger background noise. However, the peak values can still be extracted from the noisy background. The conclusion is that the HT technique is immune to noise, since even for noisy textures we can still isolate the HT peak values.

As an additional illustration, Fig. 6a shows the pattern of Fig. 5a under severe scaling (and translation). From Fig. 6a it is evident that there is still enough information present for correct classification. Figure 6b is the HT of Fig. 6a. We observe that basically only two lines were detected in each orientation. The feature vectors of the textures of Figs. 5a and 6a were calculated using the proposed classification algorithm and were found to be:

$$\text{for Fig. 5a, } \mathbf{x} = [2, 0.014, 0.290]$$

$$\text{for Fig. 6a, } \mathbf{x} = [2, 0.019, 0.283].$$

We observe that the two views of the same texture yield very similar feature vectors.

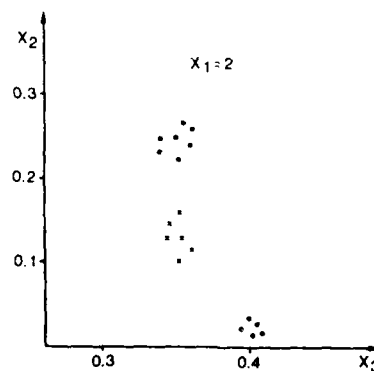


FIG. 7. Scattering diagram of three 2-directional test patterns.

For classification, the feature vector of each texture is computed following the steps of the classification algorithm. The peak values in the transform were isolated using both simple thresholding and median filtering. Both approaches performed well, but the median filter was slightly superior. The important result to examine is whether different views of the same texture yield the same feature vector, and if different textures yield different feature vectors. Since the first element of the feature vector is the number of orientations in the texture, then textures with different numbers of orientations will certainly yield dissimilar feature vectors.

Experiments performed showed that different views of the same texture always yield almost the same feature vector. The critical part was what would happen with

TABLE 1
Classification Algorithm Applied on Two Views
of the Same Texture Pattern

θ	52°	90°	128°	θ	72°	110°	148°
d	71	90	71	d	49	66	55
	70	0	69		51	66	54
	0	0	72		51	0	53
					0	0	53
					0	0	57

Normalize distances

52°	90°	128°	72°	110°	148°
0.79	1	0.79	0.74	1	0.83
0.78	0	0.77	0.77	1	0.82
0	0	0.80	0.77	0	0.80
			0	0	0.80
			0	0	0.86

Average distances

52°	90°	128°	72°	110°	148°
0.79	1	0.78	0.76	1	0.82

Shift circularly

90°	128°	52°	110°	148°	72°
1	0.79	0.78	1	0.82	0.76

Normalize angles

0°	38°	142°	0°	38°	142°
1	0.79	0.78	1	0.82	0.76

Normalize angles to 180°

0	0.21	0.79	0	0.21	0.79
1	0.79	0.78	1	0.82	0.76

Calculate variances

$$x = \{3, 0.12, 0.41\}$$

Form feature vector

$$x = \{3, 0.12, 0.41\}$$

different textures with the same number of orientations. Experiments showed that there was sufficient dissimilarity in the feature vectors for correct classification. Figure 7 is a scattering diagram of three textures, all with two orientations. Two of them have lines intersecting at 90° and the third one has lines intersecting at 79° . We notice that the first two cluster around the same value of x_3 (angle variance), at different values of x_2 (distance variance). From the scattering diagram we can see that the clusters are well separated in the feature space.

Table 1 depicts the application of the classification algorithm to two different views of the same texture. The first entry in the table is the result of searching for peaks in the transform and recording the angle at which they occur and the distance between them. We notice that the second view was rotated by 20° with respect to the first, and also that the second was taken at a smaller scale, since more peaks were found in each angle. The next two steps are to normalize all the distances by dividing by the maximum one and to average the normalized distances in each angle. Next we circularly shift so that the maximum average (one in this case) will be at the leftmost position. Then we subtract (modulo- 180°) the first angle from all the others and normalize all angles to 180° . After these steps we note that the entries in the table are almost identical. Then we form the feature vectors by calculating the variance (here the standard deviation was used) of the normalized distances and angles. Note that the first element of the feature vector is $x_1 = 3$ since three directions were detected in the texture.

SUMMARY AND CONCLUSION

In this paper a new structural approach for texture classification has been introduced. The texture primitives used for classification are the location and the relative orientation of lines in the texture. The Hough transform is used for line detection. This technique is computationally very efficient and can be applied for shapes other than lines, such as circles, parabolas, ellipses etc. If some preprocessing is applied on the texture so that wide lines become thin, then the method can be applied with excellent classification results. The feature extraction algorithm proposed is independent of geometrical transformations such as translation, rotation, and scaling. Several experimental results on idealized and natural textures were presented. Also, experiments performed on noisy textures showed that correct classification is possible even in noisy environments.

In conclusion, the Hough transform technique is a very interesting and efficient method for texture classification based on line detection, which can be generalized to include other geometrical curves as well.

ACKNOWLEDGMENTS

The support of the PSC-CUNY Faculty Research Award and the U.S. Air Force Office of Scientific Research are gratefully acknowledged.

REFERENCES

1. D. H. Berger, Texture as a discriminant of crops on radar imagery, *IEEE Trans. Geosci. Electron* **GE-8**, No. 4, 1970, 344-348.
2. T. L. Chang, Texture analysis of digitized fingerprints for singularity detection, in *Proceedings, 5th Int. Conf. on Pattern Recognition*, 1980, pp. 478-480.
3. D. G. Hubel and T. N. Wiesel, Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex, *J. Physiol. London* **160**, 1962, 106-154.

4. P. H. Lindsay and D. A. Norman, *Human Information Processing*, Academic Press, New York, 1972.
5. A. Rosenfeld and E. B. Troy, Visual texture analysis, in *Conf. Rec. for Symposium on Feature Extraction and Selection in Pattern Recognition*, Argonne, Ill., Oct. 1970, IEEE 70-C-51C, pp. 115-124.
6. A. Rosenfeld and M. Thurston, Edge and curve detection for visual scene analysis, *IEEE Trans. Comput.* C-20, 1971, 562-569.
7. R. Sutton and E. Hall, Texture measures of automatic classification of pulmonary disease, *IEEE Trans. Comput.* C-21, No. 1, 1972, 667-676.
8. D. H. Ballard and C. R. Brown, *Computer Vision*, Prentice-Hall, Englewood Cliffs, 1982.
9. R. M. Haralick, K. Shanmugam, and I. Denstein, Textural features for image classification, *IEEE Trans. Systems Man Cybernet.*, SMC-6, 1973, 610-621.
10. M. Galloway, Texture analysis using gray level run lengths, *Comput. Graphics Image Process.* 4, 1974, 172-199.
11. J. Weszka, C. R. Dyer, and A. Rosenfeld, A comparative study of textures measures for terrain classification, *IEEE Trans. Systems Man Cybernet.* SMC-6, 1976, 269-285.
12. R. Bajcsy, Computer identification of visual surfaces, *Comput. Graphics Image Process.* 2, 1973, 118-130.
13. S. W. Zucker, Toward a model of texture, *Comput. Graphics Image Process.* 5, 1976, 190-202.
14. L. Carlucci, A Formal System for Texture Languages, *Pattern Recognit.* 4, 1972, 53-72.
15. S. Y. Lu and K. S. Fu, A syntactic approach to texture analysis, *Comput. Graphics Image Process.* 7, 1978, 303-330.
16. R. W. Ehrich and J. P. Foith, A view of texture topology and texture description, *Comput. Graphics Image Process.* 8, 1978, 174-202.
17. R. W. Connors and C. A. Harlow, Towards a structural textural analyzer based on statistical methods, *Comput. Graphics Image Process.* 12, 1980, 224-256.
18. B. P. Kjell and C. R. Dyer, Edge separation and orientation texture measures, *Conf. Rec. Pattern Recog. Conf. 1985*, pp. 306-311.
19. F. M. Vilnrotter, R. Nevatia, and K. E. Price, Structural analysis of natural textures, *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-8, 1986, 76-89.
20. M. D. Levine, *Vision in Man and Machine*, McGraw-Hill, New York, 1985.
21. R. M. Haralick, Statistical and structural approaches to texture, *Proc. IEEE* 67, 1979, 786-804.
22. P. Brodatz, *Textures*, Dover, New York, 1966.
23. P. V. C. Hough, *Method and Means for Recognizing Complex Patterns*, U.S. Patent 3,069,654, Dec. 18, 1962.
24. R. O. Duda and P. E. Hart, Use of the Hough transformation to detect lines and curves in pictures, *Commun. Assoc. Comput. Mach.* 15, 1972, 11-15.
25. C. Kimme, D. Ballard, and J. Sklansky, Finding circles by an array of accumulators, *Commun. Assoc. Comput. Mach.* 18, 1975, 120-122.
26. J. Sklansky, On the Hough technique for curve detection, *IEEE Trans. Comput.* C-24, 1978, 923-926.
27. S. D. Shapiro, Transformations for the computer detection of curves in noisy pictures, *Comput. Graphics Image Process.* 4, 1975, 328-338.
28. J. Radon, *Ber. Sachs. Akad. Wiss., Leipzig* 69, 1917, 262.
29. I. M. Gelfand, M. I. Graev, and N. Ya. Vilkin, *Generalized Functions*, Vol. 5, McGraw-Hill, New York, 1966.
30. S. R. Deans, *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-2, 1981, 185.
31. Brown, Curtiss, and Sher, Advanced Hough transform implementations, in *Proceedings, IJCAI, 1983*.
32. G. Eichmann and Z. Dong, Coherent optical production of the Hough transform, *Appl. Opt.* 22, No. 6, 1982.
33. G. R. Gindi and A. F. Gmitro, *Opt. Eng.* 23, 1984, 499.
34. W. H. Steier and R. K. Shori, Optical Hough transform, *International Conference on Quantum Electronics 1985*.
35. L. K. Rangasami and A. Khotanzad, A model-based method for rotation invariant texture classification, *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-8, No. 4, 1986, 472-481.
36. O. D. Faugeras and W. K. Pratt, Decorrelation methods of texture feature extraction, *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-2, 1980, 323-332.
37. T. S. Huang (Ed.), *Two-dimensional Digital Transforms and Filters*, Springer-Verlag, Berlin, 1981.

Demonstration of a picosecond optical-phase-conjugation-based residue-arithmetic computation

Yao Li, George Eichmann, Roger Dorsinville, and R. R. Alfano

Department of Electrical Engineering, The City College of The City University of New York, New York, New York 10031

Received September 2, 1987; accepted November 16, 1987.

The use of the optical phase-conjugation (OPC) process for optical residue computation is proposed. By using an OPC-based parallel switching array, various optical position-coded residue-mapping units for carry-free addition, subtraction, and multiplication operations are described. Experimental results obtained with a picosecond mode-locked Nd³⁺:YAG laser are presented to support the proposal.

A residue number system has been suggested, because of its accurate parallel-processing capability, for carry-free arithmetic, such as addition, subtraction, and multiplication, operations.¹ An electro-optical (E-O) liquid-crystal spatial light modulator has been used to obtain the required residue-processing cyclic response.² E-O position-encoded residue mapping units have been described that use either an optical waveguide coupler array or a laser-diode grid.³⁻⁶ Other alternatives, such as the holographic optical truth-table look-up, optical symbolic substitution, and optical second-harmonic-generation residue-processing techniques, have been described.⁷⁻¹⁰ In addition, various optical decimal-to-residue and residue-to-decimal conversion schemes are available.^{11,12} Optical phase conjugation (OPC) by degenerate four-wave mixing is a technique that reverses an input beam's phase and propagation directions. OPC has found many applications in optical signal and image processing.¹³ By using an ultrafast optical $\chi^{(3)}$ nonlinear material, picosecond OPC switching has been demonstrated.¹⁴ Recently various OPC-based digital optical computing elements have been proposed and demonstrated that use its speed and parallel-processing advantages.¹⁵⁻¹⁹ In this Letter an OPC device together with a spatial-position encoding scheme is described for an optical residue processor (ORP). A prototype residue-addition mapping element that consists of an OPC cell and a combination of mirror, a beam splitter, and a cylindrical lens is described. The use of this add element for optical residue subtraction and multiplication by rearranging the input and output channels is also described. Some preliminary experimental results are presented to demonstrate an ORP that uses a picosecond mode-locked Nd³⁺:YAG laser as the source.

The core of an ORP is a set of prime modulo residue-mapping elements.³ For an ORP the two position-coded integers are directed to and switched in parallel by these mapping elements. The ORP's dynamic range is the product of all the prime moduli units employed. As an example, truth tables for a mod 4 residue addition (subtraction) and a mod 5 residue multiplication are presented in Fig. 1. For addition

(subtraction) [see Figs. 1(a) and 1(b)], left (right) cyclic rotation is needed. It has been noted⁴ that, by rearranging the input and output channels, one can use a residue-add unit to perform residue subtraction and multiplication operations. Thus, for any prime modulo residue arithmetic, a residue adder is the fundamental element. To implement a residue-add truth table, an optical switching array may be employed. The residue integers are encoded as spatial positions. For example, for the mod 4 addition, the four addend (summand) numbers 0, 1, 2, and 3 are encoded into four intersecting horizontal (vertical) channels. For each of the 16 possible addition pairs, only one intersection will be addressed. To generate the add output, the 16 addressable outputs need to be grouped into 4 sum channels. Because of the cyclic shift property of residue addition [see Fig. 1(a)], except for the four output channel 3 light spots that appear along the off-main diagonal axis, the other spots, i.e., for output channels 0, 1, and 2, appear on both sides of the off-main diagonal axis and are separated by a fixed spatial constant along this off-main diagonal direction. Thus, for the output grouping, conventional shift-invariant optical elements can be used.

In Fig. 2, a schematic of an OPC mod 4 residue adder is shown. In the top part of the figure, an OPC cell with three input beams, two counterpropagating and one off axis, is depicted. The three input beams interact within the OPC material and generate an output

	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

(a)

	0	1	2	3
0	0	1	2	3
1	3	0	1	2
2	2	3	0	1
3	1	2	3	0

(b)

	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	3	1
3	0	3	1	4	2
4	0	4	3	2	1

(c)

Fig. 1. Truth tables for mod 4 residue addition (a) and subtraction (b) and for mod 5 residue multiplication (c). After certain input and/or output permutation, both (b) and (c) can be expressed in terms of (a).

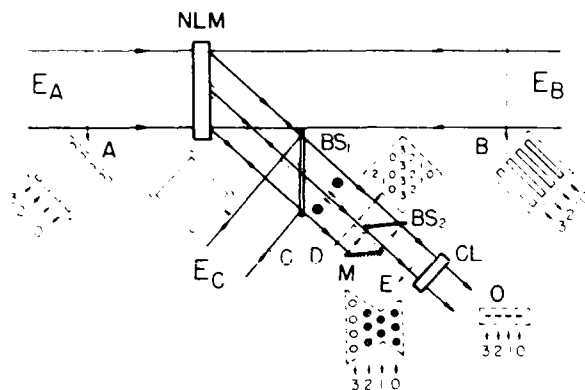


Fig. 2. A schematic of an OPC-based spatial position-coded mod 4 optical-residue addition unit. E_A , E_B , and E_C , three input beams; BS_1 , BS_2 , beam splitters; M, mirror; CL, cylindrical lens; NLM, nonlinear material. A, B, C, D, E, and O are beam profiles at various locations.

that counterpropagates with respect to the off-axis input beam. For our purpose, the two counterpropagating beams carry the two position-coded (see beam profiles A and B of Fig. 2) mod 4 inputs. The third input beam serves as an optical power supply (see beam profile C). The OPC output, which may contain as many as 16 light spots (see beam profile D), is directed by a beam splitter to a postprocessing (output grouping) unit. To group the outputs, a beam combiner that consists of a mirror-beam-splitter pair oriented at a 45° direction is inserted into the beam. This beam combiner is tuned to provide a gap that allows the four diagonal-output channel 3 spots to pass while all the other spots are properly grouped along the diagonal direction (see beam profiles D and E in Fig. 2). To generate the final addition mapping output, a cylindrical lens that spatially integrates the results along the off-main-diagonal direction (see beam profile O) is employed.

The device described above can also be used for residue-subtraction mapping. Let the additive inverse \bar{A} of a residue number A be defined as $|A + \bar{A}|_N = 0 \pmod{N}$. To subtract a residue number A , the addition of the inverse \bar{A} is performed. Thus, for example in the mod 4 case, to convert addition to subtraction the switching from the addend channel 1 (3) to the subtrahend channel 3 (1) needs to be performed. For mod N multiplication, where N is prime, a mod $N - 1$ add unit can be employed. When a homomorphic approach is used,⁴ in addition to the input (a loglike) conversion the output channels also need be permuted (an inverse loglike conversion). For example, for a mod 5 multiplication, an exchange between channels 2 and 3 for both the input and the output needs to be incorporated before a mod 4 addition element can be used.

A 32-psec pulse from a Quantel mode-locked Nd^{3+} :YAG laser is used to demonstrate the basic principles of an ORP. The experiment was divided into two parts: (1) to implement the OPC switching array and (2) to construct a mirror-beam-splitter coupler for residue-addition output grouping. First, the laser

output beam was magnified to a $1.5 \text{ cm} \times 1.5 \text{ cm}$ area from which only the central portion (about 1 cm^2) was selected for the experiments. The OPC nonlinear material was a commercially available $5.08 \text{ cm} \times 5.08 \text{ cm}$, 2-mm-thick 3-68 Corning glass filter. We understand that this filter consists of microcrystal structures of $\text{CdS}_x\text{Se}_{1-x}$ ternary compound semiconductors embedded in an amorphous glass matrix.²⁰ In addition to being fast (50 psec) (Ref. 21) and of relatively high optical nonlinearity [$\chi^{(3)} \approx 10^{-28}$ – 10^{-27} MKS units], the medium is isotropic. For the input position encoding, although one can use diode lasers and cylindrical lenses to incorporate a real-time point-to-bar shape-modulation scheme, in our preliminary experiment absorption masks were employed. Each mask slot was 1 cm long and 1.5 mm wide. To prevent spatial channel cross talk, for each mask and between every two consecutive light channels an identical-sized guard band was used. In Fig. 3, the experimental result of a 4×4 OPC switching array is shown. Two input groups (four channels each) of crossed light bars were used; the 16 light dots generated had approximately equal intensity. For a mod 4 residue adder, these outputs are then guided to a mirror-beam-splitter beam combiner. In our experiment, the beam combiner was mounted on an adjustable translation stage. The equal-splitting-ratio antireflectively coated beam splitter was 1 mm thick. The combiner was first adjusted to create a 2-mm-wide gap to allow for the passage of the four main diagonal light spots. It was then aligned to be parallel to a 45° direction and adjusted so that for the 16 input spots the output light pattern appeared to be similar to beam pattern E of Fig. 2. Finally, for one-dimensional spatial integration, a cylindrical lens was employed. With this alignment, a system verification was performed (see Fig. 4). First, for each of the four possible mod 4 addition outputs, four properly positioned light dots were obtained (see the top row of Fig. 4). The light patterns generated by using the beam combiner are shown in the middle row of Fig. 4. Note that, except for output channel 3, each of the output channels has been rearranged so that it can be located along a vertical line. Finally, in the bottom row of Fig. 4, the recorded mod 4 addition patterns made using a cylindrical lens are shown. Based on this implementation concept, other

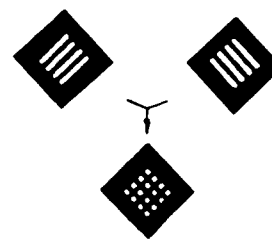


Fig. 3. Experimental result of OPC input intensity modulation. The intensity modulation (AND operation) of two input bar arrays results in a 16-light-dot output pattern that is used to generate the content of a mod 4 residue-addition truth table.

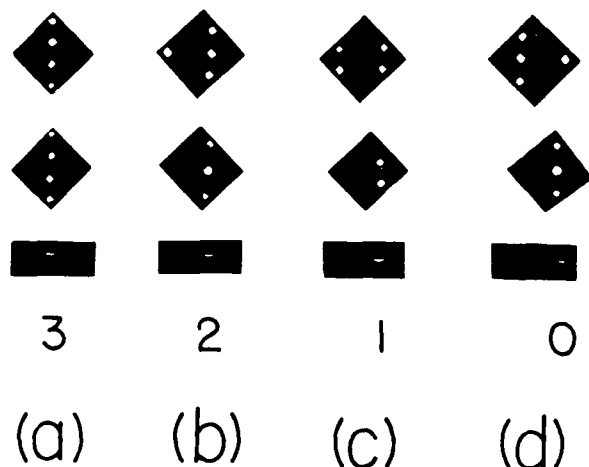


Fig. 4. Experimental results of the OPC mod 4 adder. The light patterns in the first row (corresponding to beam profile D of Fig. 2) show the four truth-table output locations for the addition results 0, 1, 2 and 3, respectively. The patterns in the second row (corresponding to Fig. 2 beam profile E) represent the corresponding beam combiner's outputs. The last row's patterns, which correspond to output 0 of Fig. 2, show the spatially integrated results.

prime modulo add units can also be realized. By using a set of parallel different modulo OPC add units together with some input and output conversion and permutation devices, an OPC-based ORP can be constructed. To reduce space-bandwidth-product problems caused by a large set of parallel mapping units, the binary coded residue (BCR) technique may be helpful. It can be shown that the dynamic range of a set of sixteen 5-bit BCR processing elements can be as high as 10^{23} . With the material and laser source employed it is difficult to achieve a high repetition rate. However, by using other high $\chi^{(3)}$ nonlinear materials with absorption peaks not in the operational range, the heating problem can be circumvented, leading to OPC-based high-repetition-rate residue processing.

The major advantages of this OPC-based ORP method are the following:

(1) It uses the degenerate wave mixing effect so that the input and output of the processor are identical-frequency optical rather than E-O hybrid signals.

(2) The use of OPC allows for an ultrafast processing speed. Although in our proof-of-principle experiments a 50-psec material was used, much faster OPC materials, such as nonlinear polymers, are available.¹⁴ Thus, compared with other existing residue-mapping schemes, this OPC-based method offers a speed advantage.

(3) The overall system uses simple optical elements, such as mirrors and beam splitters, allowing it to be miniaturized to an integrated-optics scale. With some input and output optical waveguides, the addition-mapping element can also be converted to perform subtraction and multiplication mapping. Thus

an OPC-based compact, ultrafast ORP can be constructed.

In summary, an OPC-based spatial-position-encoded ORP technique has been proposed and demonstrated. To implement a particular modulo residue-addition mapping element, an OPC setup is employed. To group the truth-table contents, a compact mirror-beam-splitter beam combiner was used. Experimental verification of a mod 4 residue-addition unit using a picosecond mode-locked Nd^{3+} :YAG laser as the source and a semiconductor-doped glass nonlinear material was described. Extension of the technique to residue subtraction and multiplication mapping was also discussed.

This research was supported in part by grants 84-0144 and 85-0212 from the U.S. Air Force Office of Scientific Research. The authors are also with the Institute for Ultrafast Spectroscopy and Lasers, The City College of the City University of New York.

References

1. N. S. Szabo, and R. I. Tanaka, *Residue Arithmetic and Its Application Computer Technology* (McGraw-Hill, New York, 1967).
2. S. A. Collins, Jr., *Proc. Soc. Photo-Opt. Instrum. Eng.* **128**, 313 (1977).
3. A. Huang, Y. Tsunoda, J. W. Goodman, and S. Ishihara, *Appl. Opt.* **18**, 149 (1979).
4. A. Tai, I. Cindrich, J. R. Fienup, and C. C. Aleksoff, *Appl. Opt.* **18**, 2812 (1979).
5. P. R. Beaudet, A. P. Goutzoulis, E. C. Malarkey, and J. C. Bradley, *Appl. Opt.* **25**, 3097 (1986).
6. A. P. Goutzoulis, D. K. Davies, and E. C. Malarkey, *Opt. Commun.* **61**, 302 (1987).
7. C. C. Guest and T. K. Gaylord, *Appl. Opt.* **19**, 1201 (1980).
8. C. D. Capps, R. A. Falk, and T. L. Houk, in *Digest of Topical Meeting on Optical Computing* (Optical Society of America, Washington, D.C., 1987), p. 58.
9. S. F. Habiby and S. A. Collins, Jr., in *Digest of Topical Meeting on Optical Computing* (Optical Society of America, Washington, D.C., 1987), p. 66.
10. G. Eichmann, Y. Li, P. P. Ho, and R. R. Alfano, *Appl. Opt.* **26**, 2726 (1987).
11. D. Psaltis and D. Casasent, *Appl. Opt.* **18**, 163 (1979).
12. C. Hannel, E. Klement, and D. Schuoecker, *Appl. Opt.* **21**, 3834 (1982).
13. T. R. O'Meara, D. M. Pepper, and J. O. White, in *Optical Phase Conjugation*, R. A. Fisher, ed. (Academic, New York, 1983), Chap. 14.
14. G. M. Carter, J. V. Hryniewicz, M. K. Thakur, Y. J. Chen, and S. E. Meyler, *Appl. Phys. Lett.* **49**, 998 (1986).
15. J. Manassah, *Opt. Lett.* **11**, 531 (1986).
16. G. Eichmann, Y. Li, and R. R. Alfano, *Appl. Opt.* **26**, 194 (1987).
17. P. C. Yeh, *Opt. Lett.* **12**, 138 (1987).
18. Y. Li, G. Eichmann, and R. R. Alfano, *Opt. Commun.* **63**, 375 (1987).
19. Y. Li, G. Eichmann, R. Dorsinville, and R. R. Alfano, "Optical digital and symbolic computation via optical phase conjugation," submitted to *Appl. Opt.*
20. R. K. Jain and R. C. Lind, *J. Opt. Soc. Am.* **73**, 647 (1983).
21. D. Cotter, *Electron. Lett.* **22**, 693 (1986).

Parallel optical pyramidal image processing

G. Eichmann, A. Kostrzewski, B. Ha, and Y. Li

Department of Electrical Engineering, City College of The City University of New York, New York, New York 10031

Received December 21, 1987; accepted March 28, 1988

Pyramidal processing is a form of multiresolution image processing in which the image is decomposed into a sequence of images at different resolutions. Pyramidal processing aims to extract and interpret significant features of an image at different resolutions. Digital pyramidal image processing, because of the large number of convolution-type operations, is time consuming. On the other hand, optical pyramidal processors, described here, are preferable in real-time image-understanding applications because of their ease in performing convolution operations. Preliminary experimental results for optical Gaussian and Laplacian pyramidal image processing are presented.

Pyramidal image processing is a form of image multi-resolution representation useful in texture analysis, biological-sample classification, motion- and stereo-image analysis, feature and shape description, image bandwidth compression, modeling of three-dimensional structures, and automatic target recognition.¹ The process of constructing a pyramidal algorithm follows the precepts of a divide-and-conquer principle. The input image is decomposed into a series of images at different resolution levels with the aim that at each level, by using sharply tuned detectors, one can identify significant features of an image. These feature detectors, at low-resolution levels, can be a line, a blob, a point, or edge detectors, while at higher-resolution levels of the pyramid they are region-level detectors. In general, the height of the pyramid is proportional to the logarithm of the image diameter. For the various applications, a large number of pyramidal algorithms are available.²

To form a pyramid, various processing techniques have been proposed. A general way to form an image pyramid is through image filtering. In this technique the higher-level, i.e., the lower-resolution, image is formed by spatial filtering followed by a decimation. The filter can be either a low-pass spatial filter, generating a Gaussian pyramid, or a bandpass spatial filter, generating a Laplacian pyramid. The difference between two successive Gaussian pyramidal images (GPI's) is a Laplacian pyramidal image (LPI). In addition to this straightforward linear filtering, to extract suitable features at each level of the GPI or LPI other linear as well as nonlinear transformations can be performed. To perform these computations for real-time image sequences, special-purpose digital image processors have been suggested and constructed. These processors must work, say, for real-time television-quality images, at a giga-floating-point operations per seconds data rate. The computational bottleneck for these processors is the large number of convolution-type operations that they must perform. This Letter reports a new pyramidal processing architecture that potentially utilizes, at each pyramidal level, real-time parallel optical spatial light modulators

and coherent Fourier optical processors. These optical processors, followed by dedicated microprocessors, are to perform the real-time feature extraction and image processing. The recent availability of commercial-quality spatial light modulators and parallel signal-processing chips promises that such a system will be both compact and inexpensive.

A GPI is a sequence of low-pass-filtered copies g_1, g_2, \dots, g_n of a primary image g_0 , where the subscript refers to the level of the pyramid. These copies are obtained by convolving g_0 with suitable low-pass filters. For a GPI, the spatial low-pass filter cutoff frequency f_c is a factor of 2 smaller than its immediate predecessor. Because of this filtering, the higher-level GPIs contain less image detail. A LPI is a set of bandpass-filtered images l_1, l_2, \dots, l_n of the image g_0 . The spatial-filter bandwidth of the LPI is an octave less than the immediate past level. Since the spatial bandwidths of the GPI's decrease by an octave, the LPI's will also have an octave spatial bandwidth decrease.

In Fig. 1 a parallel optical pyramidal image-processing system is shown. First, by using either beam splitters or a composite grating, an input image is

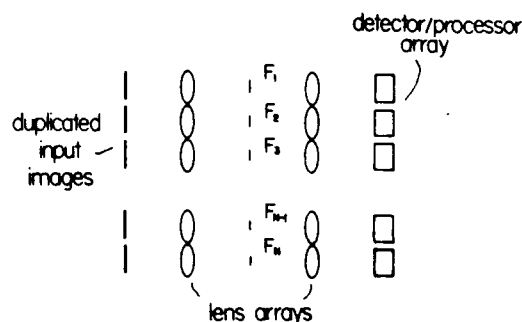


Fig. 1. Coherent optical parallel pyramidal image-processing system. N identical Fourier-transform subsystems are used in parallel. For inputs, N identical image copies are used. In the first Fourier-lens back focal plane of each subsystem, a spatial low-pass or bandpass filter F_i with $1 < i < N$ is placed. The generated GPI's (LPI's) are collected by a digital image-postprocessor array.

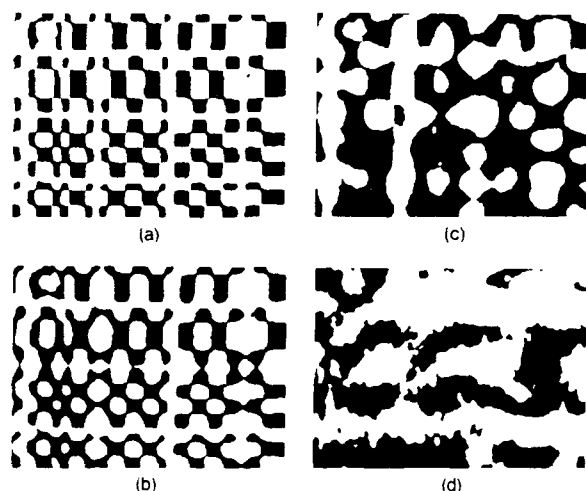


Fig. 2. First four binary GPI levels. (a) First-level image g_1 , (b) second-level image g_2 , (c) third-level image g_3 , (d) fourth-level image g_4 .

duplicated to form N parallel identical copies to be incident into N parallel optical Fourier³ processing subsystems. At each subsystem's first Fourier-lens back focal plane, a different cutoff frequency f_{Ci} , where $1 < i < N$, low-pass (bandpass) spatial filter is placed. The filtered image then passes through the second Fourier lens to generate a particular GPI (LPI). For a GPI low-pass filter, say F_i , the spatial cutoff frequency f_{Ci} is

$$f_{Ci} = D_i/2\lambda f, \quad D_i/D_{i-1} = 2, \quad (1)$$

where λ is the wavelength of light and f and D_i are the Fourier-lens focal length and the filter (pinhole) diameter, respectively. By applying the correct bandpass, i.e., a ring, filter in the back focal plane L_1 , the LPI's can be generated. These GPI's (LPI's) can be detected, thresholded, and further processed with a digital image-processor array.

It is also possible to generate a LPI by subtracting two consecutive GPI's. To achieve this task, with each subsystem a sinusoidal grating is attached to the input image so that at the filter plane two first-order diffracted-input Fourier spectra are obtained. Two GPI low-pass filters, D_i (D_{i-1}), attached with a sinusoidal (cosinusoidal) grating,⁴ will be centered on the two spectrum locations. With this method, in the subsystem output plane the two GPI's are subtracted to form a LPI. To obtain an optical LPI, other standard coherent image subtraction methods⁴ can also be used.

In our experiments a 632.8-nm He-Ne laser was used. Two pairs of identical 780-mm focal-length Fourier lenses were used. For the filter, commercial spatial filters were used. The spatial pinhole diameters ranged from 1 to 8 mm, corresponding to f_c 's from 0.98 to 7.89 line pairs/mm. For $D < 1$ mm, the diffraction effect is noticeable. By using a set of different-sized pinholes, various GPI's were recorded into the memory of a digital image-processing system. In Fig.

2(a) the g_1 is shown, while in Figs. 2(b)–2(d) the optically generated different-level GPI's are shown. These images are 200×200 pixels wide. The corresponding four f_c 's were 7.89, 3.94, 1.97, and 0.98 line pairs/mm, respectively. While the g_0 is a binary image, because of the spatial filtering the higher GPI's required additional bit planes. The higher the pyramid level, the greater is its dynamic range. After appropriate electronic thresholding, a binary GPI was obtained. For the binary GPI's, the threshold was set to 200 intensity units at the first three levels and to 240 intensity units at the fourth level. The thresholding attenuates the optical noise.

Bandpass replicas of the g_0 , the LPI's, were digitally generated by subtracting, pixel by pixel, the stored optical GPI's (Fig. 3). The size of the LPI is identical to that of its Gaussian counterpart. These LPI's may be used for further processing. To enhance the details of the LPI's, a zero-crossing method can be used.⁵ By convolving the LPI's with a 3×3 gradient mask pair F_x and F_y (Table 1), the zero-crossing copies are obtained. By shifting these masks over the LPI's, a

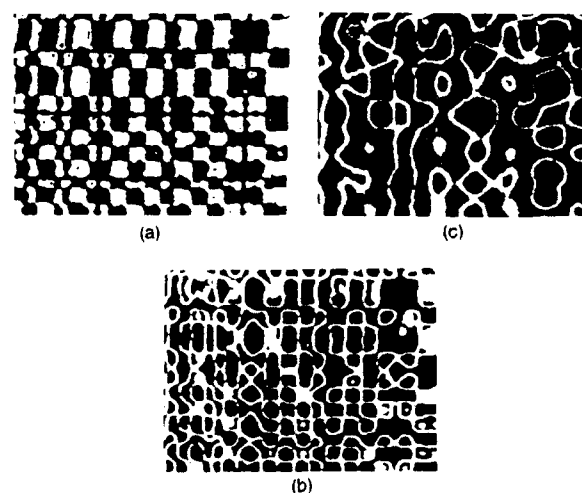


Fig. 3. First three binary LPI levels of Fig. 2. (a) First-level image l_1 , (b) second-level image l_2 , (c) third-level image l_3 .

Table 1. Gradient Masks F_x and F_y and Laplacian Mask E

F_x	1/4	0	-1/4
	1/2	0	-1/2
	1/4	0	-1/2
F_y	1/4	1/2	1/4
	0	0	0
	-1/4	-1/2	-1/4
E	0	-1	0
	-1	4	-1
	0	-1	0

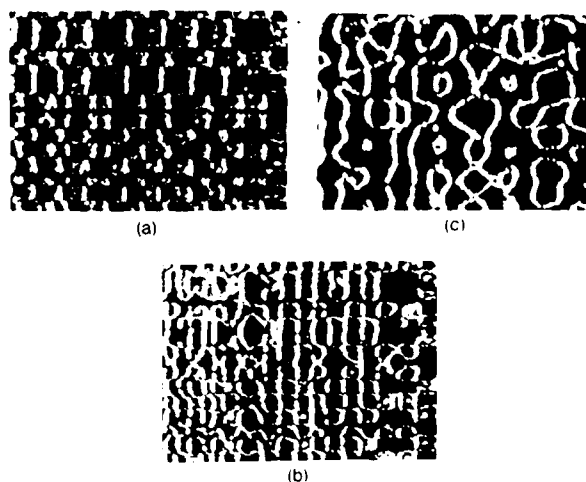


Fig. 4. First three levels of the zero-crossing LPI's. (a) First-level, (b) second level, (c) third level.

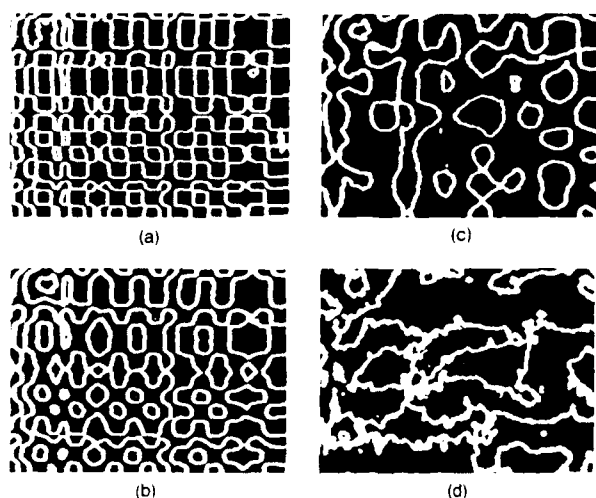


Fig. 5. Edge-enhanced copies of the binary GPI's of Fig. 2. (a) First level, (b) second level, (c) third level, (d) fourth level.

convolution of the LPI's with the F_x and F_y masks is accomplished. Figure 4 shows the LPI zero-crossing pyramid in the x direction. White (black) pixels represent positions of the image gradient that are positive to the right (left), while all other points are represent-

ed by their gray values. An alternative way to obtain the zero-crossing LPI's is through optical processing. By superimposing the Fourier transforms of F_x and F_y filters with the Gaussian low-pass filters, the GPI zero-crossing pyramid can be obtained directly. This composite optical filter can be a computer-generated hologram.

It is often necessary to extract edges from an image. Edge extraction and enhancement is widely used in image analysis and classification. Multiresolution edge enhancement can be performed by convolving a Laplacian mask E (Table 1) with either the GPI's or the LPI's.¹ Figure 5 shows an edge-enhanced GPI of Fig. 2(a). Each of the images in Fig. 5 represents multiresolution edges of g_0 at different spatial bandwidths, starting with the highest [7.89 line pairs/mm for Fig. 5(a)] and ending with the smallest [0.98 line pairs/mm for Fig. 5(d)]. Similarly to the zero-crossing pyramid, edge enhancement can also be performed optically. By inserting edge-enhanced computer-generated holographic filters, representing the Fourier transform of the Laplacian mask E , in the focal plane L_1 , multiresolution-image edge enhancement can be achieved.

To summarize, a number of optical systems for image pyramid generation are described. By performing the convolution operation by using an optical Fourier system, the optical pyramidal system has the advantage that it may attain high processing speed. By inserting a computer-generated hologram mask in the spatial-frequency plane, other pyramid processing, such as a zero-crossing LPI, may also be obtained. By using a combination of optical and digital systems, we can make full use of their respective processing capabilities.

References

1. P. J. Burt, in *Multiresolution Image Processing and Analysis*, A. Rosenfeld, ed. (Springer-Verlag, Berlin, 1984), Chap. 2.
2. A. Rosenfeld, ed., *Multiresolution Image Processing and Analysis* (Springer-Verlag, Berlin, 1984), Chaps. 3-7.
3. J. W. Goodman, *Introduction to Fourier Optics* (McGraw-Hill, New York, 1968), Chap. 7.
4. H.-K. Liu and T.-H. Chao, *Proc. Soc. Photo-Opt. Instrum. Eng.* **638**, 55 (1986).
5. J. J. Clark and P. D. Lawrence, in *Multiresolution Image Processing and Analysis*, A. Rosenfeld, ed. (Springer-Verlag, Berlin, 1984), pp. 148-168.

FAST PARALLEL OPTICAL DIGITAL MULTIPLICATION

Yao LI, George EICHMANN and R.R. ALFANO

Department of Electrical Engineering, and The Institute of Ultrafast Spectroscopy and Lasers, City College of New York, New York, NY 10031, USA

Received 19 January 1987; revised manuscript received 13 May 1987

A parallel optical binary multiplication scheme is proposed in which parallel convolution preprocessing is performed using a parallel-input optical outer-product processor together with a one dimensional either space or time integrator. Using a theta-modulation based optical A/D converter and a carry look-ahead adder array, the resulting mixed-binary partial product can be reduced to the final binary multiplication result.

1. Introduction

One of the fundamental operations in a computer is digital multiplication. The conventional approach in digital multiplication uses a shift and add scheme. To perform the digital multiplication of two N -bit numbers, after forming partial products, $N-1$ parallel adders are used. The additions can be performed in $\lceil \log_2 N \rceil$ stages, where $\lceil x \rceil$ denotes the smallest integer that is larger than x . For example, to multiply two 64-bit numbers, 63 adders in 6 parallel stages are required. It is very important to find new algorithms to perform faster multiplications. With the digital multiplication via analog convolution (DMAC) [1-4] algorithm, after performing a digital convolution on the two numbers and converting, using an A/D element, the mixed-binary partial product to its binary form, only $\lceil \log_2(N+1) \rceil - 1$ parallel adders in $\lceil \log_2(\log_2(N+1)) \rceil$ stages are needed. For example, in the $N=64$ case, after A/D conversion, only 6 adders in 3 stages are sufficient. For the multiplication of large binary numbers, in principle, the DMAC algorithm offers a faster processing speed. Since optics offers ultrafast processing speed and parallelism, optical DMAC processors have been proposed [1-4]. With a conventional optical serial DMAC processor, to perform an optical digital convolution, two acousto-optic (AO) deflectors, actuated by electronic pulse trains representing the binary serial inputs, are used. Because

of the serial input format, the convolution of two N -bit numbers requires $2N$ temporal cycles, cycles that are limited by the acoustic wave propagation speed and the AO material response time. For example, the convolution of two 16-bit numbers [4], with the currently available AO cells takes approximately 64 ns. After the convolution, to convert the mixed-binary result to binary number strings, parallel A/D converters are needed. An electro-optic (EO) interferometric A/D converter [5,6] can perform conversions in the order of nanoseconds. However, for each single N -bit EO converter, N waveguide interferometers are needed. Furthermore, because the periodic interferometric output is analog, to generate a digital number, an additional electronic comparator array must be used. Because for an N -bit serial-input digital multiplication, the DMAC algorithm requires an array of $2N-3$ A/D converters resulting in a large numbers of active EO waveguide elements.

In this communication, a new optical parallel DMAC (P-DMAC) processor is proposed. It consists of a parallel ultrafast data convolver, a fast theta-modulation EO A/D converter and an array of fast carry look-ahead adders. To increase the speed of the digital convolution preprocessing, instead of a serial, a parallel-input scheme is proposed. For the optical A/D conversion, a new theta-modulation (T-M)-based [7,8] EO device is described. This N -bit converter, that also can have a nano- or even sub-nanosecond response, requires only one active nonlinear

element. Therefore, it is more compact and less power-hungry. To add the A/D converter results, parallel carry look-ahead adder arrays are used. Using the proposed P-DMAC processor with the present technology, the overall multiplication speed is limited by the speed of $\lceil \log_2(\log_2(N+1)) \rceil$ carry look-ahead add stages. Using the existing optical ultrafast carry generation method [9] together with optical picosecond switching technology, optical fast adders may be constructed. Thus, this system can lead to faster optical binary multiplication operation.

2. A parallel optical digital convolution scheme

To perform fast convolution, rapid logic AND gates together with fast scan and sum operators are needed. It will be shown, that the first two operations can be performed via a parallel input vector outer-product processor [10]. The third, the summation, operation can be implemented via either a space or a time integrating architecture. Next, several vector outer-product-based parallel-input optical convolution devices are described.

First, a geometric optical shadow-casting-based [11,12] parallel-input optical convolution scheme is described. To optically represent the two multipliers, two superposed, spatially encoded (with logic one (zero) as a transparent (opaque) pixel, respectively) masks are utilized. As an example, consider the multiplication of the two decimal numbers $A=11$ and $B=15$. Their binary equivalents are $A=1011$ and $B=1111$. In fig. 1 (a and b), the two spatially encoded masks, representing the numbers A and B , are shown. Here, between every two consecutive bits, an opaque pixel guard bit is used. To generate the two input vector binary outer-product, these two masks are cross-overlapped. The 2-D pixel array shadowgram formed behind the overlapped masks (see fig. 1(c)) represents the two input vector outer-product. To obtain the convolution, using a cylindrical lens aligned with the shadowgram's diagonal direction, the pixel light intensities are summed. The presence of the guard bit between every two consecutive data bits prevents cross-talk between the adjacent data channels. In fig. 1(d), the light intensity pattern slightly off the lens back focal plane is shown. The number of bars in each of the seven channels

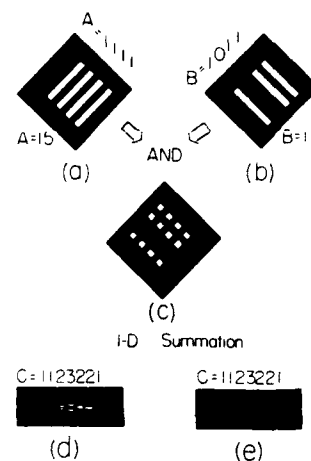


Fig. 1. Experimental result of a parallel-input optical convolution. (a) and (b) are two masks representing the two binary numbers ($A=11$ and $B=15$) to be multiplied. An opaque pixel, sandwiched between every two consecutive bits, is the guard pixel, (c) is the outer-product shadowgram formed by cross-overlapping the two masks, (d) is the result of 1-D channel-wise summation of light intensities in a plane slightly shifted from the lens focal plane, and (e) is the same result obtained right on the lens back focal plane. Both represent a mixed-binary multiplication result $C=165$.

signifies the mixed-binary product $C=1232211$. In a practical implementation, a 1-D diode detector array is placed in the lens back focal plane. In that case, instead of counting the number of bars, the focal plane intensity levels represent the mixed-binary number (see fig. 1(e)). It is interesting to note, that for a coherent illumination, this detected signal is the dc component of the two 1-D data cross-ambiguity function [13]. The side lobe of the cross-ambiguity function may be used for error detection.

While with the serial-input convolution scheme the bit-string scanning speed depends upon the speed of the acoustic wave, with the parallel-input convolution both parallel AND and self-scanning operation are performed instantaneously. Also, unlike the serial-input, where to separate the two consecutive numbers a number of idle time slots are used, with the parallel-input convolution method the data can be processed without the need for idle time. It can also be shown that, using orthogonal polarization encoding [14,16], the two parallel digital convolution channels can simultaneously be processed. Here, the

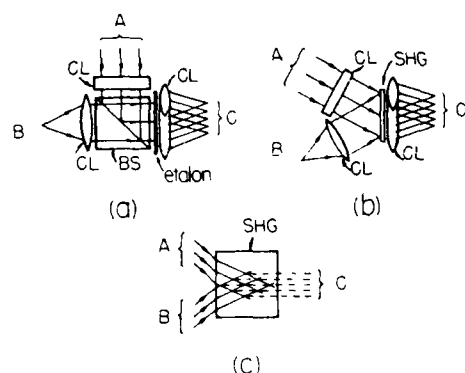


Fig. 2. Three ultrafast parallel-input optical digital convolvers. A 2-D array of (a) etalon and (b) SHG AND gates, together with two cylindrical input lenses, is used to obtain the vector outer-product of two inputs. Using a 45° oriented output cylindrical lens, the convolution result is obtained. (c) An alternative SHG parallel-input optical convolution approach where no additional lenses are required.

previously mentioned guard pixels are used as a second, orthogonal polarization encoded, channel. At the output, using a polarizing beamsplitter, the two parallel channels can be separated. To implement this scheme, either a liquid-crystal EO or a magneto-optic (MO) 2-D spatial light modulator (SLM) is needed. Currently, the processing speed of these devices is limited. To increase the processing speed, next, some possible ultrafast, parallel-input optical digital convolution schemes are proposed.

As noted earlier, to generate a parallel-input vector outer-product, each optical input digit needs to be expanded into a light bar so as to overlap the second optical input (see fig. 1(a)-(b)). The thus generated patterns can then be directed to a 2-D array of ultrafast optical AND gates. For example, in fig. 2(a) and (b), two parallel-input vector outer-product-based optical digital convolvers are illustrated. Here, either a 2-D nonlinear etalon [17] (as per fig. 2(a)) or a non-collinear second harmonic generator (SHG) AND gate array [18] (as per fig. 2(b)) is illustrated. With an etalon, the switching threshold must be set so that only when both inputs are present an output is generated. While with the SHG, using the nonlinear three wave-mixing effect, the two off-axis inputs yield an on-axis frequency-doubled AND output. In either case, to convert the digital vector

outer-product into a convolution, a 45° oriented 1-D space-integrating cylindrical lens is used. It is also possible to consider a time-integrating architecture. In fig. 2(c), as an example, a parallel time-integrating convolver is sketched. Here, to form a 2-D vector outer-product, the two parallel 3-bit inputs are injected into a thick SHG crystal. As the frequency-doubled outputs emerge, they are automatically aligned into five parallel channels. Using five time-integrating detectors, the optical digital convolution result is generated. Using any of these schemes, picosecond, parallel-input optical convolution operations can be realized [18].

3. A theta-modulation-based A/D converter

The key idea for an A/D conversion is the generation of a parallel set of different period periodic functions [5,6]. To achieve this goal, the EO interferometric approach uses a parallel set of active EO modulators. However, to A/D convert a large number, a large number of EO modulators and electronic comparators are required. It will be shown, that using a new T-M A/D converter, instead of using a large number of EO interferometric modulators and comparators, only one active and N parallel passive elements are sufficient.

The active element is a voltage controlled beam deflector that deflects a 1-D input beam to different spatial locations. There are a number of devices available to perform this function. For example, a variable grating mode SLM (VGSLM) [19] can generate, using different applied voltages, various spatial frequency grating that diffract the incident beam to different 1-D locations. The EO beam deflector [20] uses a voltage tunable index-gradient to deflect the incident light. A streak-camera [21], commonly used for ultrafast laser pulse measurement, can also be modified to be a fast beam deflector. Recently, other fast, efficient and high resolution beam deflection devices, such as the EO internal reflection deflector [22], the waveguide modulator deflector [23], etc., have also been reported. Some of these devices, because of their small capacitances (order of pF), can operate at a high (nano- or even sub-nanosecond) speed with a low (order of volts) driving voltage [23]. With these devices, the input

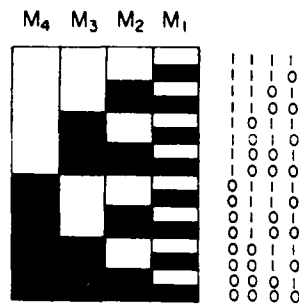


Fig. 3. Four (M_1 - M_4) binary masks to be used for a 4-bit optical T-M A/D conversion.

voltages, corresponding to detected intensity levels, are optically mapped to different spatial locations.

To convert the spatially mapped 1-D light distributions to their binary representations, a parallel set of spatially encoded masks, representing a set of different period periodic clipping operations, is used. For example, in fig. 3, for a four bit A/D conversion, four masks are shown. To illuminate the four parallel A/D conversion masks, the deflected optical beams must be focused (expanded) in the vertical (horizontal) dimension (see fig 4.) For a different horizontal-level bar, the light distribution at the mask output side represents its binary number code. Using a second cylindrical lens, the different level binary codes can be shifted to a common horizontal level where a 1-D detector array can be placed. One advantage of this new A/D conversion approach is

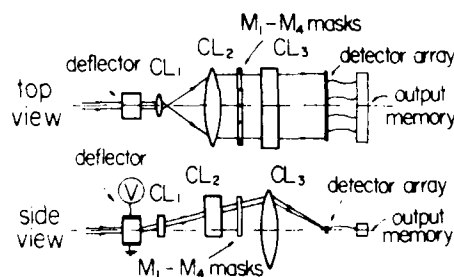


Fig. 4. Schematic diagram of an optical T-M A/D converter. (a) and (b) are top and side view of the device. The beam passing through a voltage controlled beam deflector is expanded in horizontal direction to form a deflected light beam bar to be incident on the binary masks. Using a second cylindrical lens, the converted results are shifted to a common level. At this level, a 1-D detector array is placed.

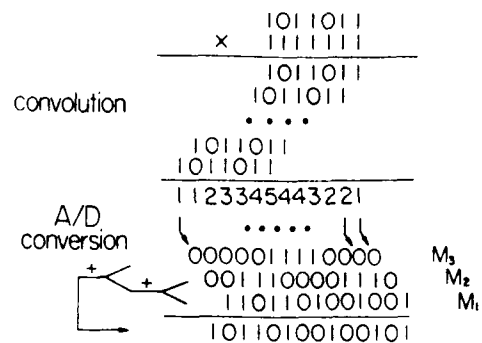


Fig. 5. A P-DMAC multiplication example in which multiplication of two 7-bit numbers is illustrated. Instead of the usual six adders in three parallel stages: this scheme uses three adders in two parallel stages.

that only a single active nonlinear element is required. Thus, in comparison with EO interferometric approach, the power consumption is drastically reduced. Also, electronic comparators are not required. Another advantage is, that by simply changing masks other binary output codes, such as Grey codes, can be obtained. Thus, this approach yields a more flexible A/D conversion scheme. With SLM generated masks, a programmable multipurpose optical A/D converter can be implemented.

4. The generation of the digital multiplication

Now that the mixed-binary number is converted to a set of binary bit strings, these results must be directed to a fast carry look-ahead adder array. Recently, a new optical carry look-ahead addition algorithm was proposed [9] where using optical multiple reflections, the carries are generated optically with a light propagation speed. With this algorithm, a complete N -bit carry look-ahead addition needs only four operational cycles. Thus, using a set of cascable ultrafast parallel optical logic switches [24], the implementation of a sub-nanosecond optical carry look-ahead adder can be expected.

As a numerical example, in fig. 5, the multiplication of two 7-bit numbers, $A=1011011$ and $B=1111111$, is illustrated. The convolution of the two bit strings yields the mixed-binary partial result $C=1123345443221$. Since the maximum weight is

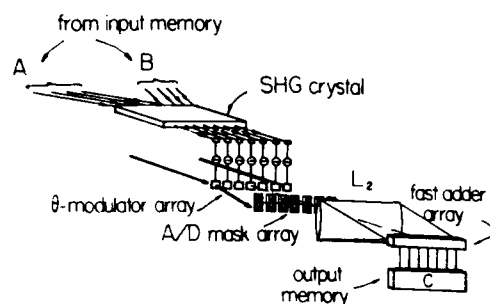


Fig. 6. A proposed real-time parallel-input 4-bit optical digital multiplication system. To generate a mixed-binary convolution result, two spatially coded 4-bit parallel inputs are directed to a SHG crystal. Using a set of beam deflectors and fast time-integrating detectors, the detected voltages are mapped to various 1-D spatial locations. Using A/D conversion masks, their corresponding binary codes are obtained. To form the final multiplication result, the generated binary signals are directed into a fast carry look-ahead adder array.

less than or equal to a seven, an array of 3-bit optical A/D converters are required. In the middle part of fig. 5, the converted results are shown. When this result is properly grouped, only two parallel addition stages are required. With these stages, the final multiplication result $C = 1011010010010$ will be generated.

In fig. 6, a proposed real-time 4-bit optical digital multiplier is shown. Here, a SHG-based 4-bit parallel-input optical convolver is used to perform an ultrafast, optical digital convolution. At each convolution output channel, the mixed-binary result is separately detected. The detected voltage signals are then used to modulate a beam deflector array. To convert the deflected beams to their binary representations, an array of A/D conversion masks are used. Finally, to generate the digital multiplication result, the partial results are directed to a fast carry look-ahead adder array.

As mentioned earlier, for the multiplication of two N -bit binary numbers, $[\log_2(N+1)] - 1$ additions in $[\log_2(\log_2(N+1))]$ stages are needed. With the DMAC scheme, the overall multiplication time is $T_C + T_{A/D} + [\log_2(\log_2(N+1))]T_A$, where the subscripts C, A/D and A denote the convolver, the A/D converter, and the adder, respectively. Compared to the conventional multiplication scheme, the time needed for the last, the addition, part is drastically

reduced. Compared to the serial-input DMAC scheme, this P-DMAC saves the convolution preprocessing time. Using the proposed parallel vector outer-product-based optical convolver, an EO-based waveguide T-M A/D converter and a fast optical carry look-ahead adder array, the digital multiplication of two 32-bit numbers in the order of nanoseconds should be possible.

5. Summary

In this communication, a parallel-input optical digital multiplication scheme has been described. For the parallel-input digital multiplication preprocessing, various optical vector outer-product processors are utilized. With either a nonlinear etalon- or SHG-based approach, the ultrafast parallel-input digital convolution can also be contemplated. To convert the convolution result from a mixed-binary to a binary form, a new optical T-M A/D converter together with a fast carry-look-ahead adder array is described. The optical T-M A/D converter uses only one active, a fast voltage controlled beam deflector, element and N passive spatially encoded binary masks. The A/D converted results are then added, using a fast carry look-ahead adder array, to generate the final multiplication result. The major advantages of this optical parallel digital multiplication scheme are (i) in comparison with its serial-input counterpart, the speed of the parallel-input convolver increases, (ii) the use of new T-M A/D converter reduces the number of active nonlinear elements leading to a more compact, less power hungry and a more economical A/D conversion, (iii) as compared to a direct multiplication, using this approach, both the number of adders and their required cascading stages are reduced leading to an overall faster digital multiplication operation. The problem that still exists with this scheme is that since the DMAC processor uses analog signals, very high accuracy optical systems for generating both outer-product and A/D conversion are needed. Recently, it has been indicated that the dynamic range and accuracy play a crucial role in determining the analog processor performance [25]. An optical analog processor is vulnerable to noisy inputs. Using high quality optical elements, uniform input beam illumination

together with high dynamic range optical detectors, processing accuracy can be enhanced.

Acknowledgement

Constructive comments from the referee are appreciated. This work is supported by a grant from the Air Force Office of Scientific Research #84-0144.

References

- [1] H.J. Whitehouse and J. Speiser, in: Aspects of signal processing with emphasis on underwater acoustics, Vol 2, ed. G. Tacconi (Reidel, Hingham, MA, 1977).
- [2] D. Psaltis, D. Casasent, D. Neft and M. Carlotto, Proc. SPIE, 232 (1980) 251.
- [3] P.S. Guilfoyle, Opt. Eng. 23 (1984) 20.
- [4] A.P. Goutzoulis, Appl. Optics 23 (1984) 4095.
- [5] H.F. Taylor, M.J. Taylor and P.W. Bauer, Appl. Phys. Lett. 32 (1978) 559.
- [6] R.A. Becker, C.E. Woodward, F.J. Leonberger and R.C. Williamson, Proc. IEEE 72 (1984) 802.
- [7] A.W. Lohmann, Optics Comm. 22 (1977) 165.
- [8] H. Bartelt, A.W. Lohmann and E.E. Sicre, J. Opt. Soc. Am. A 1 (1984) 944.
- [9] V. Chandran, T.F. Krile and J.F. Walkup, Appl. Optics 25 (1986) 2272.
- [10] R.A. Athale and J.N. Lee, Proc. IEEE 72 (1984) 931.
- [11] J. Tanida and Y. Ichioka, J. Opt. Soc. Am. 73 (1983) 800.
- [12] T. Yatagai, Optics Lett. 11 (1986) 260.
- [13] R.J. Mark II, J.F. Walkup and T.E. Krile, Appl. Optics 16 (1977) 746.
- [14] A. Lohmann, Appl. Optics 25 (1986) 1594.
- [15] Y. Li and G. Eichmann, Appl. Optics 25 (1986) 2636.
- [16] K.H. Brenner, Appl. Optics 25 (1986) 3061.
- [17] T. Venkatesan, B. Wilkens, Y.H. Lee, M. Warren, G. Olbricht, H.M. Gibbs, N. Peyghambarian, J.S. Smith and A. Yariv, Appl. Phys. Lett. 48 (1986) 145.
- [18] G. Eichmann, Y. Li, P.P. Ho and R.R. Alfano, Appl. Optics in press.
- [19] P. Chavel, A.A. Sawchuk, T.C. Stand, A.R. Tanguay, Jr. and B.H. Soffer, Optics Lett. 5 (1980) 398.
- [20] V.J. Fowler and J. Schlafer, Proc. IEEE 54 (1966) 1437.
- [21] N.H. Schiller and R.R. Alfano, Laser Focus, Aug. (1982) 43.
- [22] M.B. Chang, Appl. Optics 21 (1982) 3879.
- [23] C.S. Tsai and D. Saunier, Appl. Phys. Lett. 27 (1975) 248.
- [24] S.D. Smith, A.C. Walker, B.S. Wherrett, F.A.P. Tooley, J.G.H. Mathew, M.R. Taghizadeh and I. Janossy, Appl. Optics 25 (1986) 1586.
- [25] D. Psaltis and R.A. Athale, Appl. Optics 25 (1986) 3071.

VECTOR MEDIAN FILTERS*

T. KASPARIS and G. EICHMANN

Department of Electrical Engineering, City College of the City University of New York, New York, NY 10031, U.S.A.

Received 5 May 1986

Revised 2 February 1987

Abstract. Median filters (MF) are used both to filter 'salt and pepper' noise from signals and images and in other signal processing applications. In this paper, an extension of the MF, the vector median filter (VMF), is introduced. As opposed to the MF, the VMF outputs for each window location a number of data elements. By adjusting the VMF parameters, the MF is obtained as a VMF special case. Just like the MF, the VMF filters impulses while simultaneously preserving step changes in a signal. The VMF's principal advantage is that it reduces the total stored data signal computation time while it produces visual outputs comparable to that of an MF. Deterministic and statistical properties of the VMF are examined. Computer-generated experimental results are also presented.

Zusammenfassung. Medianfilter werden dazu gebraucht, 'Salz und Pfeffer'-Rauschen von Bild- sowie anderen Signalen zu entfernen. Im vorliegenden Beitrag wird eine Erweiterung des Medianfilters untersucht, das sogenannte Vektor-Medianfilter (VMF). Im Gegensatz zum gewöhnlichen Medianfilter werden beim VMF für jede Position des Fensters mehrere Datenelemente ausgegeben. Bei geeigneter Wahl der Parameter läßt sich das gewöhnliche Medianfilter als Sonderfall des VMF beschreiben. Ebenso wie das gewöhnliche Medianfilter unterdrückt das VMF einzelne Impulse, nicht jedoch sprungförmige Änderungen eines Signals. Der grundsätzliche Vorteil des VMF besteht in der Reduktion der Gesamtrechnenzeit; das Ausgangssignal ist mit dem eines gewöhnlichen Medianfilters vergleichbar. Überprüft werden deterministische und statistische Eigenschaften des VMF. Darüber hinaus werden einige Simulationsergebnisse beschrieben.

Résumé. Les filtres médians (MF) sont utilisés aussi bien pour filtrer dans les signaux et les images du bruit de type 'poivre et sel' que pour d'autres applications en traitement du signal. Dans cet article nous proposons une extension du MF, le filtre médian vectoriel (VMF). Par opposition au MF, le VMF donne en sortie, à chaque position de la fenêtre, plusieurs valeurs de données. Par ajustement des paramètres du VMF, le MF peut être obtenu comme cas particulier. De la même façon que le MF, le VMF filtre les impulsions tout en préservant les marches d'escalier dans le signal. L'avantage principal du VMF est de réduire le temps de calcul total, tout en donnant des résultats visuellement comparables à ceux du MF. Nous examinons ici les propriétés déterministes et statistiques du VMF. Des résultats expérimentaux sur simulation sont également présentés.

Keywords. Median filter, vector median filter, root signal, trend test, monotonic region, impulse, step edge, roof-type edge.

1. Introduction

Nonlinear filters, because of their several advantages over linear filters, are used increasingly in digital signal and image processing applications. In the class of nonlinear filters, rank filters are of particular interest. A one-dimensional rank filter

slides a window along a data array. At each window position, the window elements are sorted according to their numerical value into a list. The rank filter output is that element that falls at a predetermined position within the list. If the filter selects an element at either end of the list, then it corresponds to either a MIN or a MAX rank filter [9]. Another popular rank filter, the median filter (MF) [7] assumes an odd number of data window elements and selects the 50th percentile element in

* The grant supports of the City University of New York Faculty Research Award program and the U.S. Air Force Office of Scientific Research are gratefully acknowledged.

the data window. Median filters preserve an edge in a signal while they filter out impulses whose duration is less than $\frac{1}{2}(N - 1)$, where N is the length of the window. This type of filtering is not possible with a linear filter such as a low-pass filter which filters both signal and noise components. As a result, MFs are useful in impulse noise elimination applications.

Rabiner, Sambur and Schmidt [14] have applied median filtering to speech processing. Velleman [17] has investigated the sinusoidal response of MFs. Median filtering techniques can be extended to multi-dimensional signals. Here, the window has both a size and shape. With the window centered at a particular pixel in the multi-dimensional image, the elements within the window are sorted and the median value is used to replace the center element. Huang [6] has developed a fast two-dimensional MF that is based on a histogram calculation. Pratt [13], and also Narendra [10] who also examined the real-time implementations of this scheme, have used successive one-dimensional MFs, filtering first the horizontal and then the vertical lines, to smooth a two-dimensional image. Tukey [16], who is credited to be the first to suggest median filtering as a signal processing scheme, also suggested the following smoothing procedure: do repeated median filtering on a signal until the original signal becomes a root signal. A root signal remains invariant under median filtering.

Along with the median filter, a number of median-type filters have been suggested [2, 8, 11]. The aim of these filters is either to reduce the computational complexity without too much performance degradation, or to be more effective on certain types of noise. A survey of the MF properties is available [7]. Recently, using VLSI technology, small window size MFs were fabricated on a single chip [12, 15].

In this paper, a new median-type filter, a vector median filter (VMF) is introduced. The advantage of the VMF is that, with visual performance comparable to that of the MF, it results in reduced processing time. Because it is a multi-parameter filter, with the MF as its special case, it offers a

wider choice of filtering possibilities. After introducing this new filter, its deterministic and statistical properties are examined and compared to MF properties. Results of computer simulations will also be presented.

2. Background

The median filter [7] is defined as follows: let x be an input sequence and y be the output sequence. Then, the output MF_{2n+1} at position i is

$$y_i = \text{Med}_n[x_{i-n}, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_{i+n}],$$

where $2n+1$ is the filter window size. In this definition, the filter output y_i depends on both the past and on the future input sequence values. In order to be able to take into account end effects, n elements are appended to both ends of the original data sequence. The values of the appended elements are the same as the first element value to the left and last element value to the right of the original data sequence, respectively. Under these conditions, the first and last elements of the original data sequence will not change under median filtering.

For a window size of $N = 2n + 1$ we define the following regions: a constant neighborhood is a region of at least $n + 1$ consecutive points having the same value. Monotonic regions are regions where the signal is either increasing or decreasing. An impulse, also called a spike, is at least one but no more than n nonzero consecutive points of the same value, superimposed on a constant neighborhood. If an impulse is contained within a window, since it is narrower than n , the median value of the window cannot be an impulse element, and therefore the impulse will be eliminated from the output. If a pulse is wider than $n + 1$ points, then this pulse will be preserved. Since a step can be considered as a wide pulse, it will also be preserved. Frequently, in more general signals and especially images, 'roof-type' edges are present. These type of edges are not preserved by MFs. For example an MF would symmetrically clip the peaks of a symmetric triangular wave.

In a similar fashion, the sinusoidal characteristics of the output of an MF due to a sinusoidal input are not always preserved. It has been shown [7, 17] that the small window size MF sinusoidal response has rather large sidelobes. To reduce these undesirable sidelobes, the use of a cascade of even window size MFs has been suggested. The output of an even window size MF is the average of the two center elements in the window [7]. However, this filter no longer preserves step inputs. Another way to decrease sidelobes is to take the average of two different window size MFs [7]. For example, the filter $\frac{1}{2}(MF_3 + MF_5)$ has shallower nulls than those of either MF_3 or MF_5 .

3. Vector median filters

An MF uses an odd window size and thus a center element can always be defined. For an even window size MF, however, not one but two center elements can co-exist. For an even number of window elements, then, there is a problem in defining the output element. It has been suggested [7] to use as the MF output the average of the two center elements. An alternative is to define a vector rank filter (VRF) as the filter that simultaneously outputs a number of elements, i.e. an output data vector. As in the case of a scalar rank filter (with single element output data), we can define the MIN and the MAX vector rank filters [9] as those filters where the output vector is at either end of the sorted data window. For the vector median filter (VMF), the output values are at the center of the sorted data window.

Formally, the $VMF_{N \times M}$ is defined as follows: let a data window of size N slide along a data array. At any position, the N window elements are sorted according to their numerical value. At this position, the $VMF_{N \times M}$ output is a set of M elements, where M is less than N , situated at the center of the window, i.e. an equal number of sorted elements exist on either side of the output window between the output window ends and the data window ends. For such an output to exist,

both N and M must be restricted to be either both odd or even numbers of elements. The data window then moves M units over and the procedure is repeated.

We note that for an MF the median in the data window does not depend on the sort order, i.e. the same value is obtained if we sort in either an increasing or a decreasing order. However, this is not true for the VMF. It is easy to see that if we always sort, say, in an increasing order, a monotonically increasing signal is not affected by the filter, but a decreasing signal is distorted. Therefore, the sorting direction cannot be left arbitrary. To preserve both the increasing and the decreasing signals, the sort direction is determined by examining the signal trend in the data window. The rule we adopt is as follows: for each data element we assign a tag +1, -1, or 0, depending on whether the difference between neighboring sample amplitudes is positive, negative, or zero, respectively. Now, for the elements within the window, we compute the majority of the assigned tags. This can be accomplished by simply adding the element tags. If this result is positive (for a mostly increasing signal), then we sort the elements in an increasing order. Conversely, if the result is negative (for a mostly decreasing signal), then we sort the elements in a decreasing order. Finally, if the sum is zero (for either a constant or an oscillatory signal), the sort direction can be left arbitrary.

Since the VMF has two design parameters N and M , as compared to the MF which has only one design parameter N , as well as a trend test, the VMF has a wider choice of filtering possibilities. For example, we could adaptively adjust M based on the results of the trend test. For a slowly varying signal, we can increase M to improve speed. For a constant signal for optimum speed, we can let N equal M and skip the sort. Also, fast MF sorting algorithms could be adopted for the VMF as well.

Based on its definition, the following deterministic VMF properties can be derived:

Property 1. Any monotonically increasing or decreasing signal will be preserved.

This can be seen from the fact that the elements falling within the data window are already sorted in the correct order and thus will not be disturbed. Since a step (or edge) signal is monotonically increasing or decreasing in a region around the edge, it will be preserved by a VMF.

Property 2. Any impulse narrower than $\frac{1}{2}(N - M)$ samples will be eliminated.

This is so because if an impulse falls in the window, after the sort it will be 'pushed' to one end of that window. If the impulse is narrower than $\frac{1}{2}(N - M)$ samples, then it will fill up the window at a point where none of its elements will fall in the output vector. Since none of the impulse elements falls in the output vector, the impulse will be eliminated. By the same reasoning, any pulse wider than $\frac{1}{2}(N + M)$ will pass unaffected. Therefore, the VMF has the same two fundamental properties of the MF, namely, it preserves edges while it filters out sufficiently narrow spikes. As was mentioned earlier the MF will not preserve 'roof-type' edges such as peaks of triangulars. This is also true for the VMF. Additionally, depending on the starting point, the distortion introduced by a VMF could be asymmetrical even for a symmetrical 'roof-type' edge. However, for many classes of signals this could be a tolerable distortion.

It is known that if a signal is repeatedly MF-ed it will eventually convert the signal into a root signal [5]. An upper bound for the number of filter passes required to reach a root signal is given as $\frac{1}{2}(L - 2)$, where L is the length of the signal. However, this is not a very tight bound. It has been observed that for the same signal, the minimum number of passes required to reach a root signal decreases with increasing filter window size. Even though it has not yet been proved, it has been experimentally verified that the VMF has the same property, namely if a signal is repeatedly VMF-ed, it will eventually convert into a VMF root signal.

We also note that the VMF moves M -times faster along the data array than its corresponding MF

counterpart. However, the VMF is slightly more complex to implement. Despite this additional computational complexity, the VMF results, especially for long data arrays, in a significantly shorter processing time. As with the MF, to account for end effects, $\frac{1}{2}(N - M)$ elements are appended to the beginning of the data array, each with a value equal to that of the first element. The number of elements appended at the end of the signal depends on L , where L is the length of the data

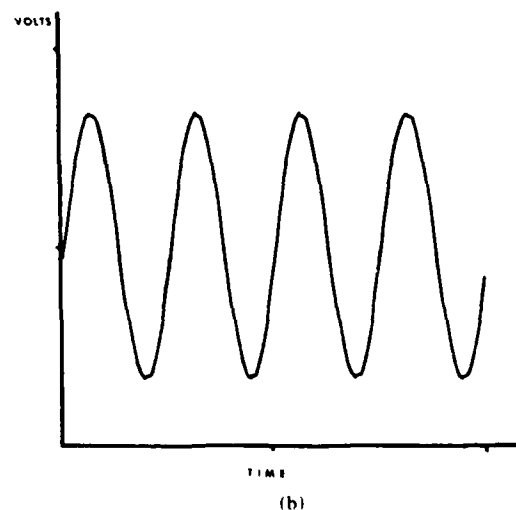
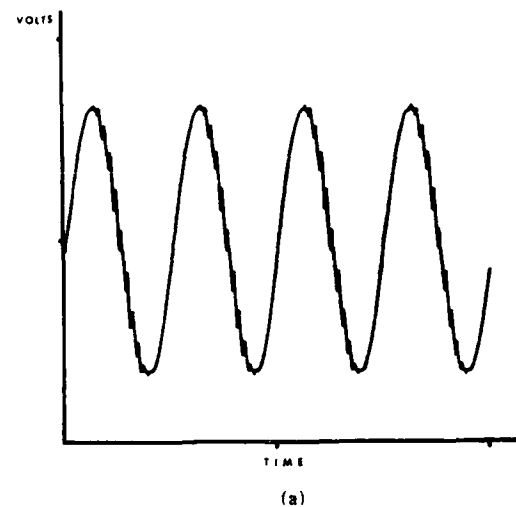


Fig. 1. (a) Output of a $VMF_{4 \times 2}$ without a trend test. (b) Output of a $VMF_{4 \times 2}$ with a trend test.

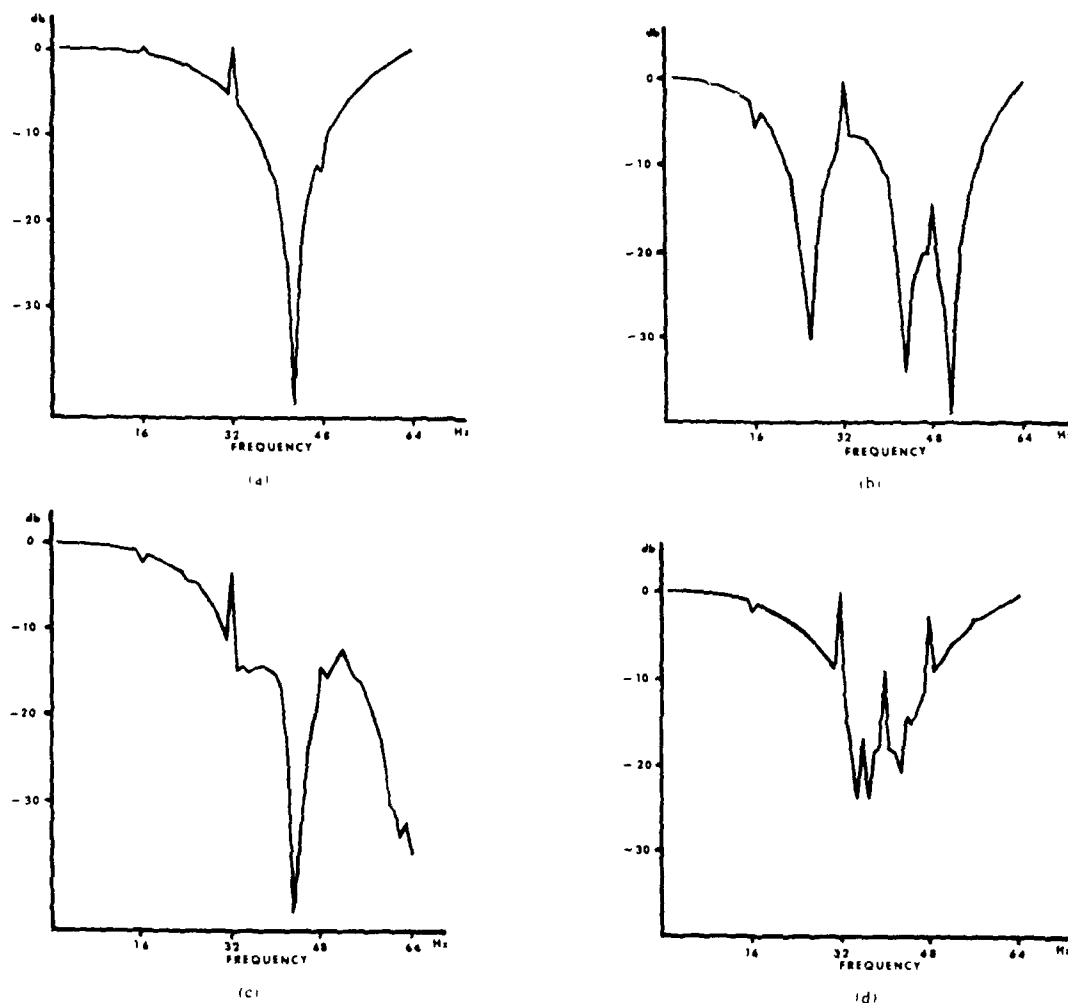


Fig. 3. (a) Gain with an MF_3 . (b) Gain of an MF_5 . (c) Gain of a $VMF_{4 \times 2}$. (d) Gain of a $VMF_{5 \times 3}$.

Fig. 2(d) displays the result of filtering by a $VMF_{14 \times 2}$. Here, all of the spikes were removed. The same result is obtained when a $VMF_{15 \times 3}$ is used. It is worth noting that in all cases the signal edges were preserved.

Next, the sinusoidal response of a VMF is investigated. The sinusoidal response of the MF has been investigated by Velleman [17]. Here, the parameter of interest is the gain, which is the ratio of the fundamental harmonic power output to the fundamental harmonic power input. Fig. 3(a)

shows the gain, on a logarithmic scale, of an MF_3 . The sinusoidal input is sampled at a rate of 128 samples per second. Fig. 3(a) shows that the MF_3 has a null of about -40 dB at a frequency of about 43 Hz. Fig. 3(b) shows the gain of an MF_5 . Here we note that there are three nulls. As a comparison, Figs. 3(c) and 3(d) show the sinusoidal response of a $VMF_{4 \times 2}$ and a $VMF_{5 \times 3}$. Comparing Fig. 3(a) with Fig. 3(d), we note that, as the window size increases, the VMF has a smoother response, concluding that for larger windows the VMF has, in

general, a smoother sinusoidal response than the corresponding MF.

4. Vector median filters: Statistical properties

Most of the work done on the statistical properties of the MF involves independent identically distributed (iid) input data. Some work was also done for non-iid data [1]. Here, we will assume that the input data is iid with probability distribution and density functions $F_x(x)$ and $f_x(x)$, respectively. Let y_i be the i th element of the VMF output at some position, where $1 \leq i \leq M$. Using a well-known order statistics result [3], the probability density of y_i will be given by:

$$f_{y_i}(y_i) = \frac{N!}{P!Q!} [F_x(y_i)]^P \cdot [1 - F_x(y_i)]^Q \cdot f_x(y_i), \quad (1)$$

where

$$P = \frac{1}{2}(N - M) + i - 1$$

and

$$Q = \frac{1}{2}(N - M) + M - i$$

In the special case where $M = i = 1$, equation (1) reduces to the output probability distribution of an MF with an output that is also iid. However, for the VMF, even though the output elements are independent, they are not identically distributed. Furthermore, the density function of each output element is one of the M functions that appears periodically with period M in the output array. Also, the mean and the variance of each output element are not the same and, thus, we must define the M -dimensional mean and variance vectors of means and variances, respectively.

As an example of such calculation, it is assumed that the input has an exponential or Laplacian probability density function (EPD). The EPD represents the noise due to laser intensity speckle. The performance of an MF on a laser speckle was examined by Frieden [4]. We will calculate the

output statistics of a VMF due to EPD inputs and compare these results with those obtained with the MF. The details can be found in Appendix A. As is shown there, the performance of the VMF on the laser speckle is slightly superior of that of the MF.

5. Computer experimental results

Using 512×512 8 bits per pixel digital images as input signals, the visual performance of the VMF as compared to that of the MF is tested. The purpose of these experiments is to demonstrate some of the visual properties of the VMF using realistic images. Fig. 4(a) shows an image with computer-generated additive uniform pseudo-random 'salt and pepper' noise (see Fig. 4(b)). The noise consists of uniformly distributed ten black or white spots per horizontal line. This noisy image is then processed with different median-type filters and their performance is evaluated both visually and by their computer run-time. A VAX 11/750 computer is used.

As was mentioned earlier 1-D filters, like the MF, were used to filter 2-D signals by first filtering the rows of the signal, and later the columns of the result. Since the VMF is basically a 1-D filter, a similar scheme can be used for filtering images. However, since the purpose of the experiments is to compare the performance of the VMF to that of the MF, for simplicity only the horizontal lines of the image were filtered. This is equivalent to filtering the sampled video signal.

Fig. 4(c)-(e) display the VMF of dimensions 3×1 , 4×2 , and 5×3 , outputs respectively. Each filter is able to remove one-sample wide impulses. Note that the $\text{VMF}_{3 \times 1}$ is a MF. While visually all three images seem identical, their computer run-times, shown in each figure, are quite different. For example, as compared to the MF_3 , the $\text{VMF}_{5 \times 3}$ requires much less processing time. Fig. 4(f), (g) shows the output of a $\text{VMF}_{5 \times 1}$ (MF_5) and $\text{VMF}_{6 \times 2}$ respectively. Because now two-samples wide impulses are removed, virtually all the impulsive



Fig. 4. (a) Source image. (b) Source image with noise added. (c) Noisy image filtered with an MF_3 . Elapsed computer time = 81 s. (d) Noisy image filtered with a $VMF_{4 \times 2}$. Elapsed computer time = 64 s. (e) Noisy image filtered with a $VMF_{5 \times 3}$. Elapsed computer time = 57 s. (f) Noisy image filtered with an MF_5 . Elapsed computer time = 104 s. (g) Noisy image filtered with a $VMF_{6 \times 2}$. Elapsed computer time = 78 s.

noise is removed. Again, inspecting Fig. 4(f), (g), we note no significant visual differences, except that the $VMF_{6 \times 2}$ is computationally faster than the MF_5 . Fig. 5 summarizes the speed improvement of the VMF in comparison to the speed of its

Signal Processing

counterpart MF. Fig. 6 presents a more detailed image example. Here, some image degradation can be observed. For example, by comparing Fig. 6(c) with Fig. 6(e), more degradation occurs as additional elements are included in the VMF. By also

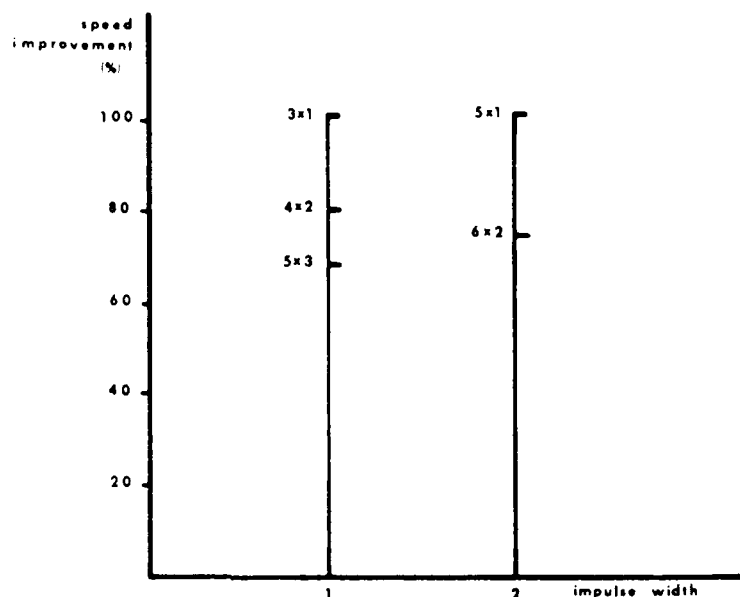


Fig. 5. Speed comparison of MF and VMF.

filtering the vertical lines, some of this degradation can be corrected. However, the difference is not very noticeable.

Similar experiments were performed on color images. Here, for each color image the same type of noise as for the black and white images was added to the three RGB color components. This noise has the appearance of different color spots in the composite NTSC image. Rather than median filtering the composite NTSC signal, a better result is obtained when the VMF-ing is performed on the RGB color components. The color VMF performance was similar to the black and white VMF image processing performance. Compared to the MF, again the VMF has the faster processing time.

6. Summary

In summary, in this paper a new median-type filter, the vector median filter, with MF as its special case, has been described. The principal advantage of the VMF over the MF is that it is

computationally faster while maintaining comparable visual performance. The speed improvement is due to faster window movement along the data array. If the signal contains extended constant regions, a further speed improvement is possible, because in this case we can eliminate the sort procedure. The sinusoidal response of a VMF filter examined and found to have a smoother response than its MF counterpart. The statistical performance of the VMF was described and, using an example, the SNR performance of the VMF and MF were compared. Experimental results on actual images were presented. With visual performance comparable to the MF, the computational speed improvement of the VMF was demonstrated. Some image degradation was observed when the output vector was increased. This is expected and is due to the deposition of some samples in the filtered signal. However, there are classes of signals like images or speech signals where some distortion is tolerable, but there would be always a trade-off between speed and degradation.

Even though there are some difficulties, it is possible to define a 2-D VMF. For example, for a



Fig. 6. (a) Original image. (b) Noisy image. (c) Output of an MF_3 . (d) Output of a $VMF_{4 \times 2}$. (e) Output of a $VMF_{4 \times 3}$. (f) Output of an MF_5 . (g) Output of a $VMF_{6 \times 2}$.

square filter window $N \times N$, we can define a 2-D output vector $M \times M$ as a square subwindow centered within the filter window. The gradient of the signal in the filter window can be used as the trend test. However, the elements of the output

Signal Processing

subwindow must be carefully selected for minimum sample deposition. In this sense, the output 2-D signal will not be generated pixel by pixel as in the case of the 2-D MF, but in $M \times M$ blocks.

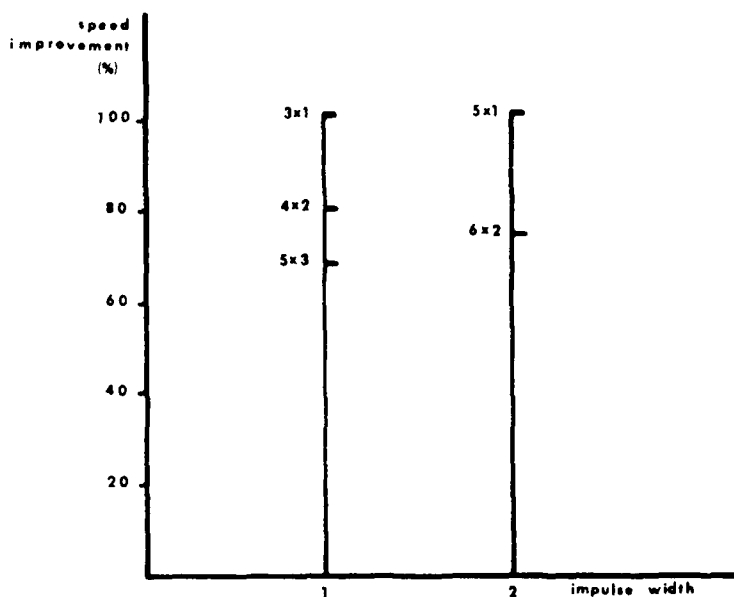


Fig. 5. Speed comparison of MF and VMF.

filtering the vertical lines, some of this degradation can be corrected. However, the difference is not very noticeable.

Similar experiments were performed on color images. Here, for each color image the same type of noise as for the black and white images was added to the three RGB color components. This noise has the appearance of different color spots in the composite NTSC image. Rather than median filtering the composite NTSC signal, a better result is obtained when the VMF-ing is performed on the RGB color components. The color VMF performance was similar to the black and white VMF image processing performance. Compared to the MF, again the VMF has the faster processing time.

6. Summary

In summary, in this paper a new median-type filter, the vector median filter, with MF as its special case, has been described. The principal advantage of the VMF over the MF is that it is

computationally faster while maintaining comparable visual performance. The speed improvement is due to faster window movement along the data array. If the signal contains extended constant regions, a further speed improvement is possible, because in this case we can eliminate the sort procedure. The sinusoidal response of a VMF filter examined and found to have a smoother response than its MF counterpart. The statistical performance of the VMF was described and, using an example, the SNR performance of the VMF and MF were compared. Experimental results on actual images were presented. With visual performance comparable to the MF, the computational speed improvement of the VMF was demonstrated. Some image degradation was observed when the output vector was increased. This is expected and is due to the deposition of some samples in the filtered signal. However, there are classes of signals like images or speech signals where some distortion is tolerable, but there would be always a trade-off between speed and degradation.

Even though there are some difficulties, it is possible to define a 2-D VMF. For example, for a



Fig. 6. (a) Original image. (b) Noisy image. (c) Output of an MF_3 . (d) Output of a $VMF_{4 \times 2}$. (e) Output of a $VMF_{5 \times 3}$. (f) Output of an MF_5 . (g) Output of a $VMF_{6 \times 2}$.

square filter window $N \times N$, we can define a 2-D output vector $M \times M$ as a square subwindow centered within the filter window. The gradient of the signal in the filter window can be used as the trend test. However, the elements of the output

subwindow must be carefully selected for minimum sample deposition. In this sense, the output 2-D signal will not be generated pixel by pixel as in the case of the 2-D MF, but in $M \times M$ blocks.

Signal Processing

Appendix A

The EPD is given as

$$f_x(x) = A e^{-x}, \quad 0 \leq x < \infty \quad (\text{A.1})$$

with

$$A = \frac{1}{p} \quad \text{and} \quad x' = \frac{x}{p^2},$$

where p is a distribution parameter. The mean value of an EPD is

$$\bar{x} = E[x] = p^2,$$

while its variance is

$$\overline{x^2} = \sigma_x^2 = p^4.$$

If we define the signal-to-noise ratio (SNR) of a random variable x as

$$\text{SNR}_x = E[x]/\sigma_x, \quad (\text{A.2})$$

then for the EPD the SNR is unity. We now assume that the random variables of (A.1) are passed once through a VMF $_{N,M}$. Substituting (A.1) into (1) yields

$$f_i(y_i) = \frac{N!}{P!Q!} A^2 \cdot [1 - e^{-y_i}]^P \cdot [e^{-y_i}]^{Q+1} \quad (\text{A.3})$$

with

$$y_i' = y_i/p^2, \quad 0 \leq y_i < \infty, \\ i = 1, 2, \dots, M.$$

For the case when $M = i = 1$, equation (A.3) reduces to the MF output distribution.

The mean value of y_i is given as

$$E_i = E[y_i] = \int_0^\infty y_i f_i(y_i) dy_i. \quad (\text{A.4})$$

Substituting (A.3) into (A.4) we find, either by direct integration or by using integration tables, that

$$E_i = p^2 \sum_{k=0}^P \left(\frac{1}{N-k} \right), \quad i = 1, 2, \dots, M, \quad (\text{A.5})$$

where, again,

$$P = \frac{1}{2}(N-M) + i - 1.$$

When $M = i = 1$ (i.e. the MF), then

$$E_1 = E[y] = p^2 \sum_{k=0}^n \left(\frac{1}{N-k} \right), \\ \text{where } n = \frac{1}{2}(N-1). \quad (\text{A.6})$$

This matches the result of Frieden [4]. We can rewrite (A.5) as

$$E_i = p^2 \sum_{k=0}^{n+m} \left(\frac{1}{N-k} \right), \\ \text{where } m = i - \frac{1}{2}(M+1). \quad (\text{A.7})$$

Comparing (A.6) and (A.7) we can see that, depending on the values of M and i , the summation in (A.7) can be extended to a greater, equal, or fewer number of terms compared to the sum in (A.6). Thus, the mean values of the elements in the output data vector can be greater, equal, or less than the mean value of the true median. Equality occurs when the median is included in the output vector.

Since the elements in the output vector do not have the same mean, we define a vector mean, i.e. a vector of means. Furthermore, we define a vector magnitude mean E as the magnitude of the vector mean, i.e.

$$E = \left[\sum_{i=1}^M E_i^2 \right]^{1/2} \quad (\text{A.8})$$

The mean value E_i can be approximated, using the Euler-Maclaurin summation formula,

$$E_i = E[y_i] \\ \approx p^2 \left\{ \ln \left[\frac{2(N+1)}{N+M-2i+2} \right] \right. \\ \left. + \frac{1}{2} \left[\frac{N-M+2i}{(N+M-2i+2)(N+1)} \right] \right\}. \quad (\text{A.9})$$

Asymptotically, when $N \rightarrow \infty$, then $E_i \approx p^2 \ln 2$, that is, all output elements have approximately the same mean as the mean of the MF (see (A.6)).

A similar calculation for the variance S_i yields

$$S_i = \sigma_{y_i}^2 = p^4 \sum_{k=0}^P \frac{1}{(N-k)^2} \quad i = 1, 2, \dots, M. \quad (\text{A.10})$$

Since the summation in (A.10) has the same form as in (A.5), similar comments to those made for the mean apply. Thus, we can define a vector variance and a magnitude variance S given by

$$S = \sum_{i=1}^M S_i. \quad (\text{A.11})$$

Equation (A.10) can be approximated by

$$S_i = \sigma_{y_i}^2 \approx p^4 \left\{ \frac{N - M + 2i + 2}{(N + M - 2i)(N + 1)} - \frac{1}{2(N + 1)^2} - \frac{2}{(N + M - 2i)} \right\}. \quad (\text{A.12})$$

Asymptotically when $N \rightarrow \infty$, $S_i = p^4/N$ for all elements, which is also the variance of the true median.

From (A.5) and (A.10), the SNR of the VMF $_{N \times M}$ can be calculated. As a figure of merit, Fig. A.1 compares the magnitude SNR of the VMF with that of the MF. As can be seen from Fig. A.1, by using the VMF, a slightly larger SNR can be achieved. Asymptotically, for larger window sizes, when $N \rightarrow \infty$, the SNR is

$$\lim_{N \rightarrow \infty} \text{SNR} = (p^2 \ln 2)/N. \quad (\text{A.13})$$

The SNR of (A.13) represents both a large window MF, and also a large data window and small output window size VMF. In this sense, asymptotically the MF and the VMF are statistically similar in performance.

References

- [1] E. Ataman and K. Wong, "Some statistical properties of median filters", *IEEE Trans. Acoust., Speech, Signal Process.*, Vol. ASSP-29, 1981, pp. 1073-1075.
- [2] A. Bovik, T. Huang and D. Munson, "A generalization of median filtering using linear combinations of order statistics", *IEEE Trans. Acoust., Speech, Signal Process.*, Vol. ASSP-31, 1983, pp. 1342-1250.

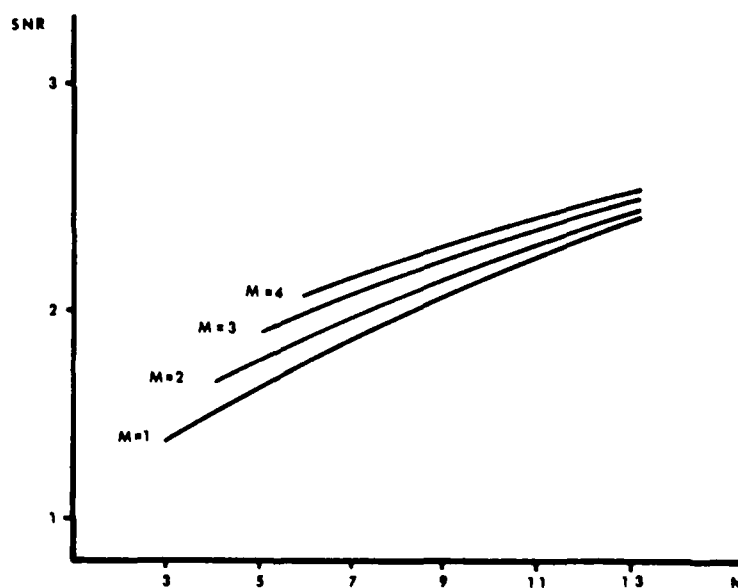


Fig. A.1. SNR of VMF and MF.

- [3] H.A. David, *Order Statistics*, Wiley, New York, 1970.
- [4] R. Frieden, "Some statistical properties of the median window", in: *Transformations in Optical Signal Processing*, *Proc. SPIE*, Vol. 373, 1981, pp. 219-224.
- [5] N. Gallagher and G. Wise, "A theoretical analysis of the properties of median filters", *IEEE Trans. Acoust., Speech, Signal Process.*, Vol. ASSP-29, 1981, pp. 1136-1141.
- [6] T.S. Huang, G.J. Yang and G.Y. Tang, "A fast two-dimensional median filtering algorithm", *IEEE Trans. Acoust., Speech, Signal Process.*, Vol. ASSP-27, 1979, pp. 13-18.
- [7] B.I. Justusson, "Median filtering: Statistical properties", in: T.S. Huang, ed., *Two-Dimensional Digital Transforms and Filters*, Springer, Berlin, 1981.
- [8] L. Ling, R. Yin and X. Wang, "Nonlinear filters for reducing spiky noises in two-dimensions", *Proc. Internat. Conf. on Acoustics, Speech and Signal Processing*, San Diego, CA, 1984, pp. 31.8.1-31.8.4.
- [9] Y. Nakagawa and A. Rosenfeld, "A note on the use of local min and max operators in digital picture processing", *IEEE Trans. Systems Man Cybernet.*, Vol. SMC-8, 1978, pp. 632-635.
- [10] P. Narendra, "A separable median filter for image noise smoothing", *IEEE Trans. Pattern Anal. Machine Intelligence*, Vol. PAMI-3, No. 1, 1981, pp. 20-29.
- [11] T. Nodas and N. Gallagher, "Median filters: Some modifications and their properties", *IEEE Trans. Acoust., Speech, Signal Process.*, Vol. ASSP-30, 1982, pp. 739-746.
- [12] K. Oflazer, "Design and implementation of a single-chip 1-D median filter", *IEEE Trans. Acoust., Speech, Signal Process.*, Vol. ASSP-31, 1983, pp. 1164-1168.
- [13] W.K. Pratt, *Digital Image Processing*, Wiley, New York, 1978.
- [14] L.R. Rabiner, M.R. Sambur and C.E. Schmidt, "Applications of a nonlinear smoothing algorithm to speech processing", *IEEE Trans. Acoust., Speech, Signal Process.*, Vol. ASSP-23, 1975, pp. 552-557.
- [15] J.A. Roskind, "A fast sort-selection filter chip with effectively linear hardware complexity", *Proc. Internat. Conf. on Acoustics, Speech and Signal Processing*, Tampa, FL, 1985, pp. 1519-1522.
- [16] J.W. Tukey, *Exploratory Data Analysis*, Addison-Wesley, Reading, MA, 1971.
- [17] P.E. Velleman, "Robust non-linear data smoothing", Tech. Rept. 89, Ser. 2, Dept. of Statistics, Princeton Univ., Princeton, NJ, 1975.

REAL-TIME OPTICAL LINE DETECTION

George EICHMANN and Yao LI

Department of Electrical Engineering, City College of CUNY, New York, N.Y. 10031, USA

Received 2 March 1987

Several alternative real-time optical Hough transform (HT) schemes that do not use moving optical elements are presented. The rotation, called for by the HT kernel, are performed by various continuous computer-generated holographic (CGH) optical elements. As opposed to a discrete CGH approaches, these implementations do not require high SBP optical elements and may lead to optical cascading lending themselves to additional Radon transform (RT) optical processing.

The detection of straight lines is an important image processing task that appears in many applications, such as target tracking, 3D object modeling, contour-following and region-growing image segmentation algorithms, etc. An important line detector is the Hough transform (HT) [1]. It has been pointed out by Deans [2] that the HT of binary images is equivalent to its forward radon transform (RT) [3], a transform that is well-known from the theory of computed tomography. The HT takes a binary image and multiplies it with an impulsive 2D kernel that contains both translation- and rotation-dependent parameters. For a line in the so-called normal form, the translation parameter p represents the shortest distance between the line to be detected and the origin while the rotation parameter θ represents the angle formed by the line and the positive cartesian x -axis (see fig. 1). For a line in the so-called intercept form, the two straight line parameters are the slope m and its y intercept b . In either case, the HT plane is a new cartesian place with the two parameters as their cartesian grid coordinates. A point in the HT plane represents a straight line from the image with its intensity as the length of the line. Collinear line segments map into the same HT plane point. Because the HT kernel is space-variant, both digital and optical HT evaluation approaches are cumbersome. The purpose of the latter section is to detail some new real-time optical HT architectures suitable for real, 2D binary images.

In a coherent optical HT implementation, Eich-

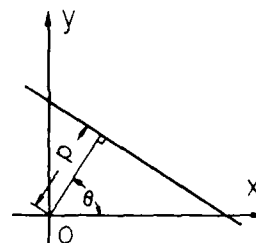


Fig. 1. The normal representation of a straight line.

mann and Dong [4] suggested that both the linear and rotation operations, called for by the HT kernel, be performed using mechanical motion. Using a standard Fourier optical system and rotating of the object plane and simultaneously linearly translating the image plane, successive constant angle lines in the HT plane can be traced out. Realizing that the impulsive kernel can also represent a rotating incoherent line source, Steier and Shori [5] used a rotating Dove prism and a linear detector array to generate at video rates the HT plane. Using a RT image processing approach, an idea that was introduced in optics by Barrett [6] and discussed by Gindi and Gmitro [7], edge-enhanced HT images were generated. While this scheme lends itself to video processing rates, it does use a mechanical rotating element. Recently, Ambs et al. [8] introduced an optical HT implementation approach that does not require a moving optical element. Instead, it has a stationary array of a mosaic of computer-generated

holograms (CGH) that form, in each small neighborhood, the HT space-variant kernel. For this system to work, it is necessary that the image does not vary in each of the neighborhoods. For an image of space-bandwidth product (SBP) N^2 , in general, a matrix of CGH of SBP N^4 is required. For realistic images, the optical SBP can be on the order of 10^{10} to 10^{12} . To write such a holographic filter, in general, an expensive E-beam generated CGH is needed. In addition, the optical elements following this high SBP hologram must also be of a high SBP quality. The purpose of this next section is to suggest some alternative CGH optical HT schemes that potentially do not require a very high SBP and also may optically cascade lending themselves available for additional optical RT processing.

To reduce the computational burden on the discrete CGH filter, the geometric operations are partitioned into a series of simpler coordinate distortions. In the optical map transformations, introduced by Bryngdahl [9], the method of stationary phase is used to alter the geometrical coordinates of the object function. The success of this method depends on the applicability of a suitable analytic phase function for the proper coordinate distortion. As opposed to the high SBP discrete CGH, and when this method is applicable, simpler continuous CGH can be utilized [10]. When the phase function is either 1D or separable 2D, the phase function can readily be determined. In some special cases, a non-separable 2D phase function can also be established.

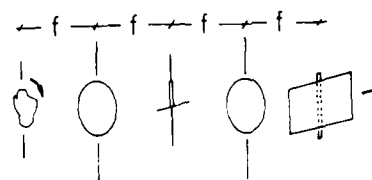
The coordinate distortion (phase) function $D(x, y)$, in a mathematical notation, is

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp[iD(x, y)] \times \exp[-iX(xu + yv)] dx dy, \quad (1)$$

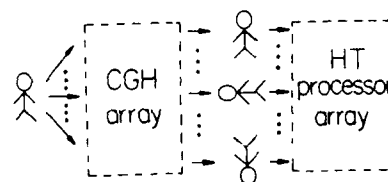
where $X = 2\pi/\lambda f$ and f is the focal length of a Fourier transform lens, λ is the wavelength of the coherent source, and (u, v) are the distorted coordinates, in a stationary phase sense, defined as

$$u = (1/X) \partial D / \partial x, \quad v = (1/X) \partial D / \partial y. \quad (2)$$

It can be shown that the mapping, from $z = x + iy$ to $w = f(z) = u + iv$, must represent an analytic function. A number of such distortion functions are available



(a)



(b)

Fig. 2. (a) An existing optical HT processing scheme that requires the rotation and translation in input and output planes, respectively. (b) A spatially parallel HT system that does not require any mechanical motions.

[11]. For example, the mapping transformation $w = z \exp(ia)$ rotates an image by a fixed angle a . The 1D coordinate distortion function $D(x) = x \ln x - x$ represents a logarithmic distortion of the object coordinate [12]. Finally, the complex 2D coordinate distortion function $D(z) = z \ln z - z$ represents a cartesian to log-radial coordinated transformation [10].

In the coherent optical HT implementation of ref. [4], the rotation and the translation motions separate in the object and image plane (see fig. 2a). Since a coherent optical Fourier transform system is translation-invariant, only the rotation operation must be represented by a CGH. In fig. 2b, one such parallel optical HT is depicted. By replacing the object with a number of copies equal to the number of desired rotation-angles and using a continuous CGH to rotate each replica by the given rotation angle, the large SBP discrete CGH optical HT system can be decomposed into a parallel set of small SBP optical HT system (see fig. 2b). Each sub-system generates a constant angle line from the HT plane. By combining the individual slices, the full optical HT plane may be dis-

played. An alternative 1D distortion scheme depends on the slope-intercept representation of the HT kernel. In this representation, the HT is

$$F(m, b) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(y - mx - b) dx dy. \quad (3)$$

We note that in this representation the kernel is in the product form in the x - and the difference form in the y -direction. To convert it into a convolutional form, in both directions, a logarithmic coordinate distortion in the two variables m and x is performed. Letting $x_1 = \ln x$ and $m_1 = \ln m$ to be the new variables, a coordinate distortion described in ref. [12], eq. (3) can be rewritten as

$$F'(m_1, b) = f'(x_1, y) * \delta(y - \exp(x_1)), \quad (4)$$

where $*$ denotes a 2D convolution operation. This logarithmic coordinate distortion allows the conversion of the shift-invariant system. In the new coordinate system, the standard Fourier transform filter system may be used. In the Fourier transform filter plane, a continuous CGH of the Fourier transform of the distorted kernel $\delta(y - \exp(x_1))$ is placed. The outputs of the system are the log of the slope and the intercept parameters. Because of the finite aperture of the optical system, however, not all the possible slope and intercept parameters may be displayed. To solve this problem, a second Fourier transform that displays the log-slope and intercept parameters relative to the x direction, i.e. using the kernel $x = ny + c$. With the two parallel Fourier transform systems, most straightline parameter values may be covered.

A final alternative, coordinate distortion scheme used a 2D cartesian to radial coordinate conversion to generate the optical HT plane. From ref. [4], an alternative representation of the HT plane is

$$F(p, \theta) = \int_{-\infty}^{\infty} F_1(w \cos \theta, w \sin \theta \exp(iwp)) dw, \quad (5)$$

where F_1 is the 2D spatial Fourier transform of the object distribution function. To evaluate this 1D Fourier transform, a cartesian to radial coordinate transformation needs to be performed. It has been indicated that with a stationary phase CGH, a cartesian to log radial transformation, i.e.

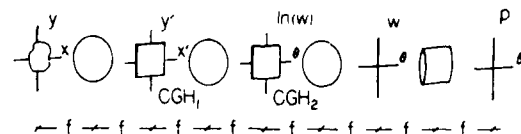


Fig. 3. An alternative rotation-and-translation-free optical HT processor.

$$w = \log[(x^2 + y^2)^{1/2}], \text{ and } \theta = \tan^{-1}(y/x), \quad (6)$$

can be implemented. By placing in the Fourier transform plane a continuous CGH consisting of a complex rotation Fourier transform filter in a stationary phase sense, in the image plane the distorted spatial Fourier transform is generated (see fig. 3). Thus, using another continuous phase CGH that transforms $\log w$ into w and a lens that provides a second 1D Fourier transform in the w and an imaging in the θ direction, the combined system will display the optical HT plane.

To summarize, a number of alternative mechanical rotation- and translation-free optical HT implementations have been proposed. Using stationary-phase computer-generated holograms, various coordinate transformations, called for optical HT implementations, are presented. With these methods, high space-bandwidth product input binary images can be processed.

This work was supported in part by a grant from the U.S. Air Force Office of Scientific research.

References

- [1] P.V.C. Hough, U.S. Patent 3 069 654 (1962).
- [2] S.R. Deans, IEEE Trans. Pattern Anal. Mach. Intell., PAMI-2 (1981) 185.
- [3] J. Radon, Ber. Saechs. Akad. Wiss. Leipzig 69 (1917) 262.
- [4] G. Eichmann and B.Z. Dong, Appl. Optics 22 (1983) 830.
- [5] W.H. Steier and R.K. Shori, Appl. Optics 25 (1986) 2734.
- [6] H.H. Barrett, Optics Lett. 7 (1982) 248.
- [7] G.R. Gindi and A.F. Bmitro, Opt. Eng. 23 (1984) 499.
- [8] P. Ambs, S.H. Lee, Q. Tian and Y. Feinman, Appl. Optics 25 (1986) 4039.
- [9] O. Bryngdahl, Optics Comm. 10 (1974) 164.
- [10] J. Cederquist and A. Tai, Appl. Optics 23 (1984) 309.
- [11] R. Sandstrom and S.H. Lee, Proc. SPIE 437 (1983) 64.
- [12] D. Casasent and C. Szczutkowski, Proc. SPIE 83 (1976) 91.

Reprinted from *Applied Optics*, Vol. 26, page 1167, April 1, 1987
Copyright © 1987 by the Optical Society of America and reprinted by permission of the copyright owner.

Compact optical generalized perfect shuffle

George Eichmann and Yao Li

CUNY-City College, Department of Electrical Engineering,
New York, New York 10031.

Received 7 October 1986.

0003-6935/87/071167-03\$02.00/0.

© 1987 Optical Society of America.

For the solutions of many scientific and engineering problems, parallel processing has been shown to be a fast way to process information. To distribute and to interconnect massive amounts of data between stages of parallel processing elements, fast and efficient interconnection networks are needed. It has been indicated that, for some applications, shuffle-exchange networks¹⁻³ are very effective in handling such data interconnections. Shuffle-exchange networks are implemented using repeated stages of the so-called perfect shuffle (PS) together with arrays of exchange boxes that can independently either exchange or bypass the adjacent lines. Different combinations of PS and exchange box arrays have found applications¹⁻³ in evaluating polynomials, in sorting data, in transposing matrices, as well as in computing the fast Fourier transform.

Given the inherent parallelism of optics, interest has been focused on developing parallel optical computing architectures, and in particular, on the implementation of optical shuffle-exchange networks. Goodman *et al.*⁴ and Marhic⁵ proposed the use of optical fibers or waveguides for an OPS. However, for large data arrays, large bundles of fibers are

needed. To take full advantage of the free-space propagation property of optical waves, Marhic,⁵ Lohmann *et al.*^{6,7} and Brenner and Huang⁸ suggested the use of unguided implementation approaches. An unguided OPS consists of either a hologram or a suitable lens and prism combination. While the holographic OPS requires monochromatic light inputs, the lens/prism-based counterpart can also be used with white light illumination. In this Letter, additional and more compact unguided OPS geometries are suggested. New transmissive and reflective OPSs are described. Finally, an implementation of an optical generalized PS (OGPS) is also discussed.

The PS $P_N(i)$ is defined as²

$$P_N(i) = (2i + [2i/N]) \bmod N \quad 0 \leq i \leq N-1, \quad (1)$$

where $N = 2^i$, i and j are integers and $[2i/N]$ represents the largest integer that is $\leq 2i/N$. When binary symbols are used as input line addresses, after a PS permutation, the binary addresses of the output lines represent a right shift operation. Using this PS cyclic shift permutation property together with arrays of exchange boxes, any address configuration can be performed into any other configuration of the order of $(\log N)^2$ steps.³

In the stretch-mask-add approach,⁷ the unguided OPS consists of four prism wedges and two positive spherical lenses with focal lengths f_1 and f_2 , respectively. Correspondingly, the total length of the system is $2f_1 + 2f_2$. To maintain the same output channel spacing as that of the input, the length f_2 must be twice the length of f_1 leading to a total optical system length of $6f_1$.

A more compact OPS, using a new unguided OPS implementation, is suggested here. A PS requires that half of the inputs diverge by a factor of 2 while they interlace with those from the second half inputs. To obtain this divergence, a negative cylindrical lens may be employed. In Fig. 1(a), a negative cylindrical lens-based OPS is shown. Here, side by side, two identical aperture (D) and focal length (f) negative lenses are used. For simplicity, the sketch shows plano-concave negative lenses, where the unused portions of the lenses are not shown. Collimated input beams illuminate the plane of the lenses, where the input mask is located. The output beams, at the back focal plane of the lenses, represent the shuffled result. For an N -bit input, using geometric optics, the bit or channel period (d) and spot size (a) are determined by

$$d = \frac{D}{N-1} \quad (2)$$

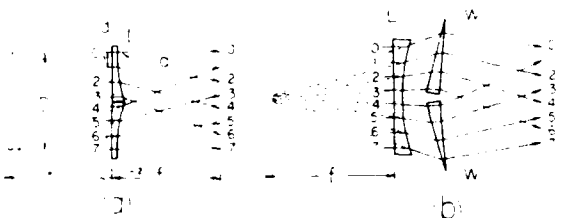


Fig. 1. Schematic diagrams of an OPS where D is the system aperture, a is the input channel size, and d is the channel spacing. (a) A lens-based system: L_1 and L_2 are two identical focal length (f) negative cylindrical lenses. (b) An alternative lens-based system where a single negative cylindrical lens together with two identical prism wedges are used.

$$a \leq \frac{D}{2(N-1)} \quad (3)$$

Because the output spot size is magnified by a factor of 2, the input spot size [Eq. (3)] is constrained to one-half of the input bit period d . For example, if the input bit spot size a and their spacing d is 0.1 and 0.25 mm, respectively, a 50×50 -mm² aperture OPS can optimally handle as many as 40,000 light channels. Compared to the stretch-mask-add approach, this system is more compact since it has only two optical elements while its length is reduced by a factor of 6. With this method, the size of the output bit is identical to the stretch-mask-add approach spot size. However, because this OPS generates a divergent output, in a shuffle-exchange network the exchange boxes must be able to recollimate the optical beam. When a two-port optical waveguide switch is used as the exchange box, by proper front and back lens adjustments the beams can be demagnified to their original sizes. This compact OPS can also be implemented with large aperture reflective optical elements, i.e., two identical, side by side, radius R convex cylindrical reflective surfaces. If necessary, the output spots can be separated out by a beam splitter. In either case, the divergence operations are performed by two identical components (lens or mirror).

It is also possible to generate the required operations with only a single negative cylindrical optical element (lens or a mirror). In Fig. 1(b) the use of a single cylindrical lens-based OPS is illustrated. First, using a single negative lens, the two divergence operations are performed. Second, using two prism wedges, each half of the diverging results is stretched and interlaced together to generate the final OPS output.

Next, an optical implementation of a generalized PS (GPS)⁹ is described. The interest in GPS stems from the fact that in many applications, instead of using $N = 2^i$ inputs, the use of other composite integer (M) inputs is required. A GPS $[G_{k,n}(i)]$ characterized by the two integers k and n ($k \geq$

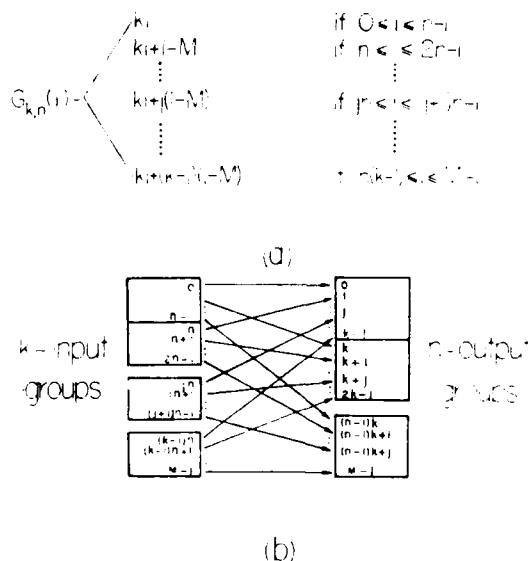


Fig. 2. Generalized PS permutation interconnection. (a) A more detailed input and output relation. (b) A graphic example showing the input and output relation. Note that both the input and output are each divided into k and n groups.

2, $n \geq 2$) such that the total number of inputs M ($M = kn$) is defined as"

$$G_{k,n}(i) = ki + [i/n](1 - M). \quad (4)$$

In Fig. 2(a), some details of the GPS permutation formula are given, while in Fig. 2(b) a corresponding permutation example is illustrated. Here, both the input and output ports are divided into k and n groups, respectively. In each of the k input groups, for example, in the j th group, there are n input lines that are to be distributed to a fixed place (the j th line as in the example) in each of the n output groups. Note that a PS $P_N(i)$ is a special case of GPS, i.e., $G_{2N,2}(i)$. Since for each of the k input groups an identical magnification divergence operation is performed, for an OGPS implementation, k pieces of either transmissive or reflective optical elements, cut from either identical focal length negative cylindrical lenses or identical radius cylindrical reflective surfaces, can be utilized. In Fig. 3, using transmissive optical elements (a negative cylindrical lenslet array), three OGPS cases, $G_{3,2}(i)$, $G_{3,3}(i)$, and $G_{4,2}(i)$, are illustrated. In general, for each of k identical size elements the aperture A is

$$A = \frac{2D}{M-1}. \quad (5)$$

Because the OGPS output is collected at a distance $(k-1)f$ measured from the input plane, compared to input, the output diverges by a factor of $k-1$. Thus, the input bit or channel size must be chosen as

$$a \leq \frac{D}{(k-1)(M-1)}. \quad (6)$$

While this method can be used for arbitrary k and n , because of the beam divergence it is only practical for relatively small k .

New unguided OPS geometries have been proposed. Using either a pair of negative cylindrical lenses or convex reflective surfaces, either transmissive or reflective OPSs can be implemented. Compared to the stretch-mask-add approach, this method uses fewer optical elements and a more compact geometry. The method can also be generalized to implement an OGPS.

This work was supported in part by grants 84-0144 and 85-0212 from the U.S. Air Force Office of Scientific Research.

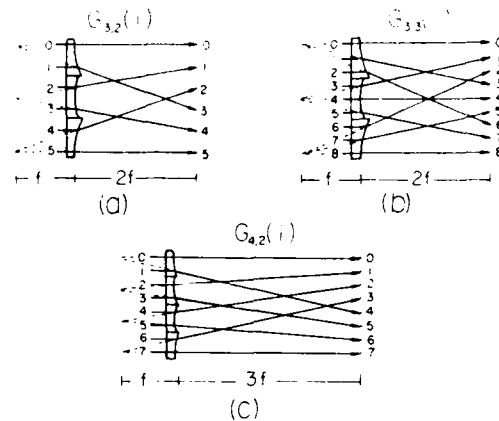


Fig. 3. Three lens-based OGPS implementation examples: (a) $G_{3,2}(i)$; (b) $G_{3,3}(i)$; (c) $G_{4,2}(i)$.

References

1. H. S. Stone, "Parallel Processing with the Perfect Shuffle," IEEE Trans. Comput. **C-20**, 153 (1971).
2. D. S. Parker, Jr., "Notes on Shuffle/Exchange Type Switching Networks," IEEE Trans. Comput. **C-29**, 213 (1980).
3. C. L. Wu and T. Y. Feng, "The Universality of the Shuffle-Exchange Network," IEEE Trans. Comput. **C-30**, 324 (1981).
4. J. W. Goodman, J. F. Leonberger, S. Y. Kung, and R. A. Athale, "Optical Interconnections for VLSI Systems," Proc. IEEE **72**, 850 (1984).
5. M. E. Marhic, "Combinatorial Star Couplers for Single-Mode Optical Fibers," FOC-LAN **84**, 175 (1984).
6. A. W. Lohmann, "What Classical Optics Can Do for the Digital Optical Computer," Appl. Opt. **25**, 1543 (1986).
7. A. W. Lohmann, W. Stork, and G. Stucke, "Optical Perfect Shuffle," Appl. Opt. **25**, 1530 (1986).
8. K.-H. Brenner and A. Huang, "Optical Implementations of Symbolic Substitution," J. Opt. Soc. Am. **A 1**, 1292 (1984).
9. J. Mikloško, "Correlation of Algorithms, Software and Hardware of Parallel Computers," in *Algorithms, Software and Hardware of Parallel Computers*, J. Mikloško and V. E. Kotov, Eds. (Springer-Verlag, Berlin, 1984).

Parallel optical logic using optical phase conjugation

George Eichmann, Yao Li, and R. R. Alfano

City University of New York, City College, New York, New York 10031.

Received 3 July 1986.

0003-6935/87/020194-03\$02.00/0.

© 1987 Optical Society of America.

Nonlinear optical phase conjugation (OPC) offers solutions to many problems in real-time optical signal and image processing. In an analog mode, using OPC optical signals (images) can be processed in parallel.¹ Also, using OPC, digital logic implementation has been suggested.² The optical logic variables are represented by either the beam on/off or its orthogonal polarization states. Using these representations, a number of binary optical logic elements such as AND, EOR, NOT, have been described. In this Letter, a new method to perform parallel digital optical logic that combines OPC with parallel logic generation techniques is proposed. This technique is suitable to implement optically all sixteen binary logic operations.

To perform parallel optical logic, Bartelt *et al.*³ proposed coherent theta modulation, while Ichioka and Tanida⁴ (I-T) and Yatagai⁵ suggested incoherent geometric optical shadow-casting (OSC) methods. In the following, the use of both OSC schemes is discussed. In both the I-T and Yatagai approaches the logic encoding is identical, with the difference being in how the different logic operations are performed. While I-T uses different LED source patterns, Yatagai uses a switchable operation mask to obtain different logic operations. In both methods, the optical beams must pass three consecutive, either a source and two input or two

input and a output mask, planes. This triple-multiplication implies an optical triple-product operation. To perform an optical triple-product operation, one may use acoustooptic Bragg cells,⁶ nonlinear third harmonic generation,⁷ as well as the nonlinear OPC.⁸ Here, we only discuss an OPC parallel optical logic processing method.

An OPC technique based on the use of Yatagai's scheme is described first. In Fig. 1, a typical OPC experimental setup is shown. Three input beams generated from the same laser, labeled E_A , E_B , and E_C , are collimated into a cubic [$\chi^{(3)}$] nonlinear material (NLM). The beams E_A and E_B are mutually phase-conjugated. The third beam E_C serves as the probe. The nonlinear interaction of the three beams in the NLM generates a polarization source that radiates a fourth beam:

$$E_O \propto \chi^3 E_A \cdot E_B \cdot E_C^*, \quad (1)$$

where the * stands for complex conjugation. In analogy to Yatagai's parallel logic geometry, the two encoded logic input masks T_A and T_B (the encoding and operation schemes for both Yatagai's and I-T's methods are summarized in Table I) are inserted into the path of beams E_A and E_B , respectively, while the operation mask T_C is placed on the E_C beam. The phase-conjugate signal E_O , separated out by a beam splitter, is the logic output. This output beam possesses the same properties as Yatagai's arrangement. Since this OPC geometry is no longer collinear, both input and output beams can be separated either spatially or directionally. This separation allows for the optical interconnection of various stages of parallel logic processors. These processors are needed to perform multiple-instruction multiple-data (MIMD) parallel processing. Also, to generate a phase-conjugate signal, as long as E_A and E_B beams counter-

Table 1 Optical Encoding (Black/White Code) Techniques for Either the Yatagai- or I-T-Type Optical Parallel Logic Processing

coded inputs		logic function	O ₀	O ₁	O ₂	O ₃	O ₄	O ₅	O ₆	O ₇	O ₈	O ₉	O ₁₀	O ₁₁	O ₁₂	O ₁₃	O ₁₄	O ₁₅
A	B		0	A·B	A·B̄	A	Ā·B	B	A⊕B	A+B	A+B̄	A⊕B̄	B̄	A+B̄	Ā	A+B	Ā·B̄	1
		Yatagai operation mask																
		I-T input LED pattern																

Both methods use identical input logic variable encoding (see left side). To implement different logic operations, coded operation masks (for Yatagai's method) and coded input LED patterns (for the I-T method) are shown on the right.

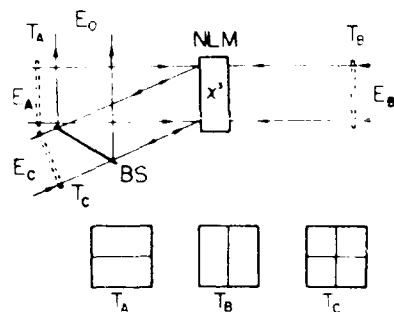


Fig. 1. OPC implementation of a Yatagai-type parallel logic processor. NLM, cubic nonlinear material; BS, beam splitter; E_A , E_B , and E_C , collimated input beams; E_C , the phase-conjugate output beam; T_A and T_B , two coded input masks; and T_C , the logic operation mask.

propagate (phase-conjugated), the third E_C beam can be incident from any angle. Thus, using different operation masks with various angular probe beams together with an angular multiplexer that selects at a given time a different probe beam, both space- and angle-variant optical parallel processing of large amounts of data are possible. Furthermore, since the OPC is a coherent optical technique, the combination of the OPC parallel digital logic method and other standard coherent optical analog processing techniques can make it a more flexible arrangement.

Next, using the NLM cell as a real-time triple-product operator, an I-T-type parallel OPC logic generation is discussed. As mentioned earlier, to obtain the various optical binary logic operations, the I-T method uses an array of switchable LED source patterns. From a geometric point of view, the interlaced output pattern due to the different LEDs can be interpreted as an optical shadowgram. However, this operation is also equivalent to a 2-D optical multiplication followed by an incoherent correlation. It is well known that 2-D coherent optical correlation can be performed using a Fourier transform lens. Based on this concept, in Fig. 2 an coherent real-time OPC correlator for implementing I-T-type parallel logic operations is shown. This coherent OPC correlator was first proposed and demonstrated by White and Yariv¹⁰ as a means to perform various coherent analog image convolution and correlation operations. In addition to a NLM cell, three equal focal length Fourier transform lenses are also employed. For the 2-D optical signals, E_A , E_B , and E_C , in the front focal planes of three lenses, the phase-conjugated output is

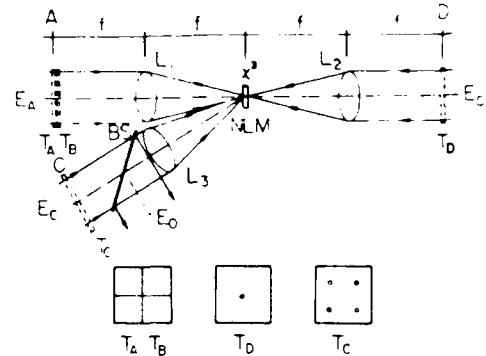


Fig. 2. OPC implementation of an I-T-type parallel logic processor. L_1 , L_2 , and L_3 , three equal-focal-length Fourier transform lenses; BS, beam splitter; NLM, nonlinear material; E_A , E_B , and E_C , collimated input beams; E_C , output beam; T_A and T_B , two coded input masks superposed in the plane A; T_C , a mask containing a central δ -function placed in the plane D; T_C , a logic operation mask placed in the plane C.



Fig. 3. Correlation of the overlapped inputs with four shifted δ -functions. D , width of coded input variable mask; d_{x1} and d_{x2} , displacements of the i th δ -function; (left) overlapped inputs containing four possible illuminated areas (see the left side of Table I); (middle) four shifted δ -functions; (right) the correlation result containing nine possible illuminated areas.

$$E_{C'} = \chi^{-1} E_A \otimes E_B \star E_C, \quad (2)$$

where \otimes and \star denote correlation and convolution operations, respectively. To obtain the required multiplication operation for the I-T-type parallel OPC logic, the two logic input masks T_A and T_B are superimposed and placed on the E_A beam. In analogy to the I-T LED source array, a corresponding source mask T_C is inserted into the path of the E_C beam with four transparent dots representing four displaced Dirac δ -functions. Since no additional convolution is required, the E_C beam mask has a single on-axis dot representing a central δ -function. To obtain the correct correlation function, the E_C beam δ -function displacements d_{x1} and d_{x2} , where $i = 1, 2, 3$, and 4, must be chosen as

$$|d_x| = |d_y| = D/4, \quad (3)$$

where D is the input pixel size. Since the convolution of a function with a δ -function shifts that function after the OPC correlation/convolution operation, the needed I-T-type parallel OPC logic is obtained. To insure the correct correlation/convolution result, the NLM cell should be made thin enough to enclose only the optical Fourier spectra of the three interacting beams.⁸⁻⁹ For clarity, in Fig. 3, the 2-D correlation of a square and four properly displaced δ -functions is illustrated. The leftmost box contains four subsquares, each of which represents a possible transparent area. When all four middle box δ -functions are on, the correlation result, shown in the rightmost box, contains nine possible illuminated areas. With this method, using sixteen possible on/off combinations of the four δ -functions, the sixteen two-variable binary logic operations can be performed. In the I-T method, because the input and output areas are not identical, in general, it is difficult to cascade two or more of these processors. For this reason, a conventional (black/white encoded) I-T OSC method is classified as single-instruction multiple-data (SIMD) processing.⁴ To perform parallel MIMD operations, another encoding scheme, such as the use of polarization encoding,¹⁰ is needed. In polarization encoding, the two orthogonal linear polarizations are used as two binary states. The thus encoded logic inputs, after a particular OSC manipulation, can generate two sets of orthogonal output patterns representing two different logic operations.

To summarize: the use of a real-time OPC triple-product device to generate coherent optical parallel logic operations is described. A NLM can be used as a major interconnection device that connects logic inputs to different output ports where different logic operations can be performed. The use

of both the Yatagai and I-T-type parallel OPC logic implementation schemes is discussed. When both input signals are generated in real time, i.e., by two spatial light modulators, fast real-time parallel logic processing of 2-D data can be performed.

This work was supported in part by a grant from the Air Force Office of Scientific Research No. 84-0144.

References

1. A. E. Chiou and P. Yeh, "Parallel Image Subtraction Using a Phase-Conjugate Michelson Interferometer," *Opt. Lett.* **11**, 306 (1986).
2. T. R. O'Meara, D. M. Pepper, and J. O. White, in *Optical Phase Conjugation*, R. A. Fisher, Ed. (Academic, New York, 1983).
3. H. Bartelt, A. W. Lohmann, and E. E. Sirc, "Optical Logical Processing in Parallel with Theta Modulation," *J. Opt. Soc. Am.* **A 1**, 944 (1984).
4. Y. Ichioka and J. Tanida, "Optical Parallel Logic Gates Using a Shadow-Casting System for Optical Digital Computing," *Proc. IEEE* **72**, 787 (1984).
5. T. Yatagai, "Optical Space-Variant Logic-Gate Array Based on Spatial Encoding Technique," *Opt. Lett.* **11**, 260 (1986).
6. W. T. Rhodes, "Acousto-optic Signal Processing: Convolution and Correlation," *Proc. IEEE* **69**, 65 (1981).
7. B. Winitzer, "Measurement of Ultrashort Laser Pulses," *Opt. Commun.* **48**, 225 (1983).
8. J. O. White and A. Yariv, "Real-Time Image Processing via Four-Wave Mixing in a Photorefractive Medium," *Appl. Phys. Lett.* **37**, 5 (1980).
9. A. Yariv and R. A. Fisher, in *Optical Phase Conjugation*, R. A. Fisher, Ed. (Academic, New York, 1983).
10. Y. Li, G. Eichmann, and R. R. Alfano, "Optical Computing Using Hybrid Encoded Shadow Casting," *Appl. Opt.* **25**, 3636 (1986).

AN AND OPERATION-BASED OPTICAL SYMBOLIC SUBSTITUTION PATTERN RECOGNIZER

Yao LI, George EICHMANN, Roger DORSINVILLE and R.R. ALFANO

*Department of Electrical Engineering, and The Institute for Ultrafast Spectroscopy and Lasers
The City College of The City University of New York, New York, NY 10031, USA*

Received 9 March 1987

A new optical binary pixel pattern recognizer for optical symbolic substitution (OSS) digital computation is proposed. Using optical spatial shift and optical phase-conjugate AND operations, input symbolic pattern recognition can be performed. Some initial experimental results obtained with picosecond laser pulses are presented.

1. Introduction

Optics offers both the speed and parallelism required for digital signal processing and computation. Two promising parallel optical computation schemes are a parallel theta-modulation-based logic system suggested by Bartelt, Lohmann and Sirci [1] and optical shadow casting logic processors proposed by Ichioka and Tanida [2], and Yatagai [3]. To generate a complete set of two-variable boolean logic operations, there are other possible optical parallel structures [4-7]. Most recently, Brenner, Huang, and Streibl [8] proposed an optical symbolic substitution (OSS) computation scheme. With an OSS scheme instead of decomposing the computation into stages of boolean logic operations that use multiple inputs to generate a single output, both multiple spatial inputs and their relative locations are utilized to generate, in parallel, multiple spatial outputs.

The OSS method can be decomposed into a pattern recognition and scription step [8]. In its operation, pattern recognition (searching for the dark pixel locations) consists of possible input multiple spatial shifts, a collinear superposition (an OR), a threshold NOR, and a masking (an AND) operations. In this approach, for the shift and superposition operation, an interferometer is employed, while for the NOR operations, a matrix of parallel nonlinear optical threshold NOR gates is also used. From the DeMorgan's theorem, however,

$$(A + B + \dots + X + Y + Z) = A B \dots X Y Z \quad (1)$$

a multiple-input NOR gate can be synthesized with INVERTERS and AND gates [8,9]. For INVERSION, instead of searching for the dark, the white (transparent) pixels are recognized. Compared to a threshold NOR, an optical threshold AND-based approach has the advantage that it is easier to implement optically. However, in terms of signal-to-noise ratio, an optical threshold-AND gate may introduce an additional recognition error. When an N pixel pattern is to be recognized using an optical threshold AND gate, an output of one will be achieved only when the detected total intensity ($N I$, where I is a single pixel intensity) is above a threshold. Thus, with this type of gate, one must distinguish between levels $(N-1)I$ and $N I$. As N increases, its noise immunity decreases. For this reason, a threshold NOR logic-based approach possesses a larger signal-to-noise ratio than its threshold AND-based counterpart [8]. For the pattern scription step, the previously recognized pixel pattern is used in another device where only spatial shift operations are performed.

In this letter, a new OSS pattern recognizer that employs a multiple-input boolean AND element is described and demonstrated. To recognize a multiple-white-pixel pattern, in addition to multiple spatial shifts, only AND operations are used. To prevent noise accumulation caused by a threshold-based approach, an optical phase-conjugate (OPC) mul-

Table 1

Fourteen possible four-pixel optical symbolic patterns to be recognized. According to the number of transparent pixels, these patterns can further be classified into three groups. To recognize a N ($N=1,2,3$) transparent-pixel pattern, $N-1$ relative spatial shifts and a $N+1$ input AND gate is needed

types	input patterns	No. of shifts	No. of inputs for AND
A		0	2
B		1	3
C		2	4

multiple-input boolean AND element is employed [7,10-11]. The advantages of using an OPC-based symbolic recognition scheme are discussed. Some preliminary experimental results using picosecond laser pulses are presented.

2. An AND-based symbolic pattern recognition

For an OSS operation, the first step is a symbolic pattern recognition. The input is a 2-D rectangular pattern array that contains several elemental light pixel patterns (the elemental patterns). For the purpose of this discussion, let the elemental pattern consist of a square of four-pixels. When the modulation is a transparent/opaque code, this elemental pattern can form sixteen different pixel combinations. Excluding the two trivial patterns (either all transparent or opaque) that can be recognized by other methods, in table 1, the remaining fourteen combinations are listed. These patterns can be classified into three groups: A, B and C. Since for the recognition of the four group-A patterns only an optical masking operation on these patterns is needed, no further discussion is presented.

To recognize the six group-B patterns, shift operations must be performed. As an example, consider the input pattern shown in fig. 1. The input image contains four four-pixel elemental patterns where one of them that contains two transparent main-diagonal pixels is to be searched. To recognize this four-pixel elemental-pattern, first, the image is replicated into two parts which are then either spatially shifted up or to the left by one unit, respectively. Together with

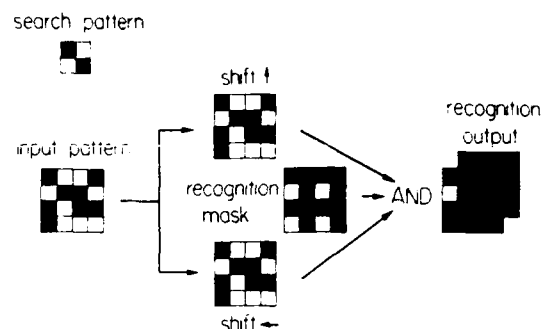


Fig. 1. Example of a four-pixel type-B pattern recognition. To locate the search pattern, two copies of the spatially shifted input are directed, together with a recognition mask, to a three-input parallel AND device.

a recognition mask that consists of four transparent pixels at the four elemental pattern's lower left-hand corners, the shifted images are next directed to a three-input 2-D parallel AND gate. In this case, its output indicates that the search pattern resides at the upper-right input image location. The three-input 2D AND operation can also be viewed as two cascaded two-input AND operations, i.e. an AND between the two shifted inputs, and a second AND between the first AND output and the recognition mask. In particular, the two two-input AND operations may help to discriminate against both intra- and inter-elemental-pattern noises. For other type-B inputs (see table 1), differential spatial shifts and recognition mask are used. When one of the two replicated inputs is stationary, only a single spatial shift, a shift that allows the two transparent pixels to overlap, is sufficient. For example, in fig. 1, by fixing the lower image position, only the upper image needs to be shifted in the upper right direction to a position where the two intra-elemental-pattern transparent pixels overlap. In general, to recognize a two-transparent-pixel pattern, a single relative shift and a three-input AND operation are required.

Similarly, for the group-C patterns, to discriminate against the intra-elemental-pattern noises, three copies of an input image with two relative shifts and a three-input AND element are needed. To discriminate against inter-elemental-pattern noises, an additional masking (AND) operation is used. As an example, in fig. 2, a type-C pattern recognition is

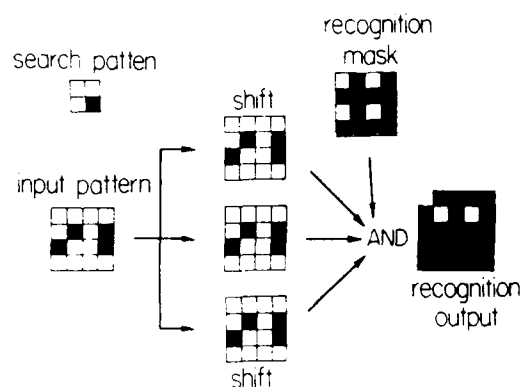


Fig. 2. Example of a four-pixel type-C pattern recognition. To locate the search pattern, three copies of the spatially shifted input are directed, together with a recognition mask, to a four-input parallel AND device.

shown. The input image contains two elemental search patterns. Here, either a single four-input or three two-input parallel AND elements needs to be employed. In general, to search for a N -transparent-pixel elemental pattern, $N-1$ pattern spatial shifts and a N -input parallel AND element must be used.

3. Optical implementations

In this section, the use of an OPC effect for a boolean multiple-input AND-based OSS pattern recognizer is described. In an OPC material, two counterpropagating optical beams and a third (coming from an arbitrary direction) arrive at a third order optical nonlinear $\chi^{(3)}$ material. When the phase matching condition is satisfied, in a backward (with respect to the third beam) propagation direction, a fourth, the so-called optical phase-conjugate (OPC) beam, is generated [10]. Using this OPC beam generation property, various analog and digital signal processing and computation applications have been proposed [7,10-11].

The OPC device can be considered as a three-input boolean logic AND element. In fig. 3(a), an OPC-based AND device is shown. A collimated type-B input image beam is divided, using a beamsplitter, into two copies. Directed by two plane mirrors, the two beams counterpropagate, with a relative spatial

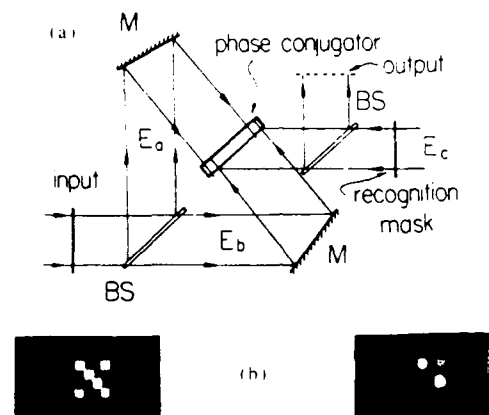


Fig. 3. (a) A schematic ultrafast OPC-based symbolic pattern recognition device. (b) Experimental result obtained with a 32 ps Nd:YAG laser source and a 2 mm thick CS₂ cell. The input pattern that consists of four elemental-patterns is shown on the left-hand side where the elemental-pattern to be searched contains two main-diagonal transparent light cells. The recognized output pattern is shown on the right-hand side.

shift, to an OPC material. A third beam, containing the recognition mask, is also directed to the $\chi^{(3)}$ material. The generated OPC signal counterpropagates with respect to the third beam. Finally, using a second beamsplitter, this signal is directed to the system output. With a slight modification, the OPC device can also be configured as a four-input, an element called for the type-C pattern recognition, AND element. In this case, all the three OPC input ports are used to carry spatially shifted input images. At the output port, the recognition mask is placed.

Using a polarization encoding method, it is also possible to collinearly combine the third beam with one of the counterpropagating inputs [10] (see fig. 4 for the geometry). Assume that the two counterpropagating inputs are linearly polarized. With a polarizing beamsplitting cube, the third input beam that is orthogonally polarized is also collinearly guided, with one of the counterpropagating inputs, to the nonlinear material. In this case, the polarization of the OPC output is identical to the third input polarization direction and it can easily be separated by the polarizing beamsplitter.

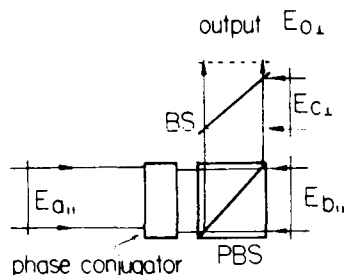


Fig. 4. An alternative OPC-based symbolic pattern recognizer. With orthogonal polarizations, input $E_{a,i}$ and $E_{b,i}$ are collinearly directed into the OPC material.

4. Experiment

To verify the operation of the OPC-OSS pattern recognizer, using a QUANTEK modelocked Nd^{3+} :YAG laser that generates 32-ps optical pulses, an experiment was performed. A $2\times$ telescope was employed to expand the spatial profile to an area of 2 cm^2 out of which a small portion (about 1 cm^2) was used. For a larger aperture OPC pixel pattern recognition, before beam expansion, the source needs to be spatially filtered. In the experiment (see fig. 3(a) for the geometry), the sixteen-pixel input image contains four four-pixel type-B elemental patterns. The search pattern was a main-diagonal transparent pixel elemental pattern. With an appropriate spatial shift, the two beams (A and B) containing two shifted copies of the input mask were directed from the opposite directions to a 2 mm thick CS_2 cell. The recognition mask used in the probe beam (C) was angularly shifted by 5° from one of the counterpropagating beams. As illustrated in fig. 3(b), the picosecond OPC output signals shows that the expected search pattern was located at the input image upper-left and lower-right hand corners. The residue at the upper-right corner is the stray light noise. Using a threshold detector, this stray light noise can be filtered.

5. Discussion

This new OPC-based symbolic recognition scheme has the following advantages over the other schemes [8,9]:

(1) Instead of performing, as required by the scheme of ref. [8], three different (an image superposition equivalent to a logic OR, a threshold NOR and a masking equivalent to an AND) logic operations, here, only a single logic element, a multiple-input optical AND gate, is employed.

(2) The OPC-based scheme allows ultrafast processing. Using materials such as semiconductor-doped glasses or nonlinear polymers pico- or sub-picosecond OPC switching response times have been observed [12,13]. When the input binary pixel pattern is also generated by an ultrafast 2D modulation scheme such as from a parallel bistable etalon array [14] an ultrafast OSS pattern recognition can be performed.

(3) The OPC-based approach reduces the cumulative error that occurs with a threshold-based AND gate. This is true because the generation of an OPC-AND output is based on the input phase-matching condition that does not, to the first order, depend on the input intensity levels.

(4) The OPC-based approach also reduces the interference errors that occur in a collinear input pattern superposition geometry. With the refs. [8,9] schemes, it is important to perform a large aperture nearly perfect image superposition. Otherwise, any disturbance that changes input wavefront by a fraction of a wavelength will produce a slowly changing interference pattern leading to recognition errors. This is not the case with the OPC AND-based device since the off-axis angular inputs produce much higher density interference fringes. The averaged pixel intensity of the high density fringes can reduce the decision error.

(5) The OPC outputs are potentially cascable. With a material, e.g. a multiple-quantum-well semiconductor, that exhibits a large nonlinearity and with an increased beam interaction region, e.g. a collinearly combined polarization-encoded OPC geometry (see fig. 4), an amplified OPC output can be obtained. The OPC amplification has been experimentally observed in CS_2 [10]. Thus, multiple-stages of OSS operation are possible.

One of the problems with the OPC-based scheme is that to recognize a N -transparent-input (where N is larger than three) pattern, a number of cascading AND stages are needed. This sequential operation does decrease the recognition speed. One way to

minimize this problem is to use a tree-type (in $\log_2 N$ steps) logic decomposition structure. Another problem with the OPC-based scheme is that the OPC off-axis input is scaled causing a vignetted output. The polarization-encoded counterpropagating OPC geometry (see fig. 4) can eliminate this problem.

6. Summary

A new OPC binary pixel pattern recognizer for OSS has been proposed and demonstrated. Using a number of spatial shift and AND operations, a given optical pixel pattern can be recognized. For an optical implementation, mirrors and beamsplitters were used to obtain the required spatial shifts while an OPC-based device was used for the logic AND operation. Using an OPC-based scheme, ultrafast symbolic pattern recognition was experimentally demonstrated.

Acknowledgement

This work is supported by a grant from the Air Force Office of Scientific Research #84-0144.

References

- [1] H. Bartelt, A.W. Lohmann and E.F. Siere, *J. Opt. Soc. Am. A1*, (1984) 944.
- [2] Y. Ichioka and J. Tanida, *Proc. IEEE* 72 (1984) 787.
- [3] T. Yatagai, *Optics Lett.* 11 (1986) 260.
- [4] M.T. Fatehi, K.C. Wasmundt and S.A. Collins, Jr., *Appl. Optics* 20 (1981) 2250.
- [5] G. Eichmann, Y. Li and R.R. Alfano, *Opt. Eng.* 25 (1986) 91.
- [6] T. Minemoto, S. Numata and K. Miyamoto, *Appl. Optics* 25 (1986) 948.
- [7] G. Eichmann, Y. Li and R.R. Alfano, *Appl. Optics* 26 (1987) 196.
- [8] K.H. Brenner, A. Huang and N. Streibl, *Appl. Optics* 25 (1986) 3054.
- [9] M.T. Tsao, L. Wang, R. Jin, R.W. Sprague, G. Giglio, H.-M. Kulcke, Y.D. Li, H.M. Chou, H.M. Gibbs and N. Peyghambarian, *Opt. Eng.* 27 (1987) 41.
- [10] A. Yariv and P. Yeh, *Optical waves in crystals* (John Wiley and Sons, New York, 1984).
- [11] T.R. O'Meara, D.M. Pepper and J.O. White, in: *Optical phase conjugation*, ed. R.A. Fisher (Academic Press, New York, 1983) p. 537, ch. 14.
- [12] G.M. Carter, J.V. Hryniewicz, M.K. Thakur, Y.J. Chen and S.E. Meyler, *Appl. Phys. Lett.* 49 (1986) 998.
- [13] P.P. Ho, N.L. Yang, T. Jimbo, Q.Z. Wang and R.R. Alfano, Ultrafast resonant optical Kerr effect in 4BCMU PDA, to appear in *J. Opt. Soc. Am. B*.
- [14] T. Venkatesan, B. Wilkens, Y.H. Lee, M. Marren, G. Olbright, H.M. Gibbs, N. Peyghambarian, J.S. Smith and A. Yariv, *Appl. Phys. Lett.* 48 (1986) 145.

MULTISTABLE FABRY-PEROT RESONATOR WITH AN ACTIVE SAGNAC INTERFEROMETER AS ITS RETRO-REFLECTOR

Yao LI, George EICHMANN and R.R. ALFANO

Department of Electrical Engineering, and the Institute for Ultrafast Spectroscopy and Lasers, City College of The City University of New York, New York, NY 10031, USA

Received 7 July 1986; revised manuscript received 12 August 1986

Multistable operations of an active Fabry-Perot (FP) resonator with an active Sagnac interferometer as its optically pumped variable reflectance retro-reflector (SIFP), is analyzed. Intensity-dependent transmission (reflection) curves are derived. The corresponding multistable operation is also indicated. By adjusting the SIFP parameters, it can be configured to be either an optical limiter, or an amplifier, or a logic or a memory element.

1. Introduction

Recently, the dynamic properties of a Sagnac, also known as a cyclic or an antiresonant ring, interferometer (SI) have been studied. A passive SI is a highly mechanically autostabilized device [1] that can be used as an optical retro-reflector. Using the SI as an equivalent end reflector, both mode-locked Nd:glass and dye lasers have been constructed [2,3]. Among its applications are laser cavity dumping [4], optical nonlinear parameter measurements [5], optical switching [5,6], and optical digital computing [7].

To study optical multistability, there are two major analytic approaches. In the first approach, Maxwell's equations together with proper boundary conditions are utilized to analyze multistability in different optical resonator geometries, such as a co-linear Fabry-Perot (FP) resonator [8], a ring resonator [9], etc. In a second approach, introduced by Felber and Marburger (F-M) [10], a replacement of a linear by an equivalent nonlinear refractive index in a non-absorbing co-linear type FP resonator is used. In this work, the use of a FP resonator with an active SI as its retro-reflector (SIFP), is proposed. In the following sections, both passive SIFP transmission and reflection equations are derived. By substituting the linear by its nonlinear intensity-dependent phase counterpart and using the F-M approach, subsequently, different aspects of an active SIFP are studied. Finally, the

use of an active SIFP to perform optical amplification, limiting, and digital logic are also described.

2. The SIFP as an equivalent FP cavity

In fig. 1(a), an SIFP with three mirrors ($M_i, i = 1, 2, 3$) and a beam-splitter (BS) and with wave direction-dependent amplitude transmission and reflection coefficients as $t_j(t_j')$ and $r_j(r_j')$ with $j = 1, b$, and with round-trip phases for the co-linear, the clockwise (counterclockwise) SI sections as $\delta_j, \delta_{s,1}(\delta_{s,2})$, respectively, is depicted. In fig. 1(b), an equivalent FP for this SIFP is shown. The purpose of this section is to derive the intensity dependent transmission (reflection) coefficients of its equivalent FP.

The SI complex amplitude transmittance and reflectance, denoted as t_s and \tilde{r}_s , are ^{*}

$$t_s = r_2 r_3 \{ r_b^2 \exp(i\Delta\delta_s, 2) - r_b^2 \exp(-i\Delta\delta_s, 2) \} \exp(i\bar{\delta}_s), \quad (1)$$

$$\tilde{r}_s = 2r_2 r_3 r_b t_b \cos(\Delta\delta_s, 2) \exp(i\bar{\delta}_s), \quad (2)$$

where the average SI counterpropagating wave phase

^{*} A similar equation has been described in ref. [1]. Here, we let two SI counterpropagating beams have different round-trip phases ($\delta_{s,1} \neq \delta_{s,2}$).

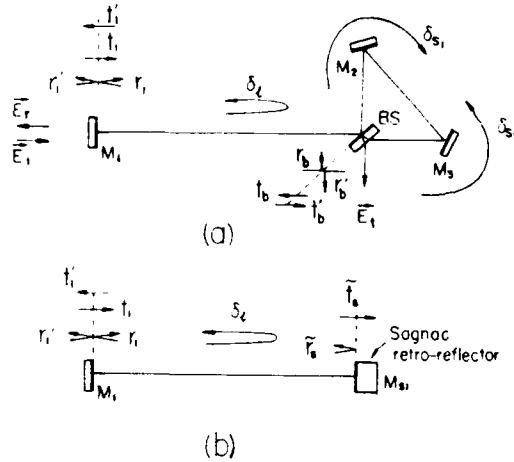


Fig. 1. (a) A schematic diagram of the SIFP: E_i and E_r (E_p), the resonator input and transmitted (reflected) output waves; M_i ($i = 1, 2, 3$), mirrors; BS, beamsplitter; t_i (r_i) and r'_i ($i = 1, 2$), direction-dependent mirror and BS wave amplitude transmittance and reflectance; δ_l and δ_{s1} (δ_{s2}), co-linear section and the SI clockwise (counterclockwise) section round-trip phases. (b) The equivalent J-P diagram with an effective SI as an retro-reflector (M_{SI}), t_s and r_s , SI wave amplitude effective transmittance and reflectance.

and phase difference are $\bar{\delta}^* = (\delta_{s1} + \delta_{s2})/2$ and $\Delta\delta_s = \delta_{s1} - \delta_{s2}$, respectively. The corresponding SI intensity transmittance (T_s) and reflectance (R_s) are

$$T_s = |t_s|^2 = R_2 R_3 - R_s, \quad (3)$$

$$R_s = |r_s|^2 = 4R_2 R_3 R_b T_b \cos^2(\Delta\delta_s/2) \\ = R_{sm} \cos^2(\Delta\delta_s/2), \quad (4)$$

where R_{sm} is the maximum SI intensity reflectance. In agreement with ref. [1] for unity intensity reflectance for both R_2 and R_3 and $R_s = T_s = 0.5$, Eqs. (3) and (4) become zero and one, respectively.

For the SIFP, the equivalent amplitude transmission (t) and reflection (r) coefficients are

$$t = \frac{E_t}{E_i} = \frac{t_1 \tilde{t}_s \exp(i\delta_l/2)}{1 - r_1 \tilde{r}_s \exp(i\delta_l)} \\ r = \frac{E_r}{E_i} = \frac{r'_1 + \tilde{r}_s \exp(i\delta_l)}{1 - r_1 \tilde{r}_s \exp(i\delta_l)} \quad (5)$$

Using eqs. (1)–(5), the SIFP intensity transmittance (T) and reflectance (R) are

$$T = |t|^2 = \frac{A}{1 + F \sin^2(\delta_T/2)}, \quad (6)$$

$$R = |r|^2 = \frac{B + F \sin^2(\delta_T/2)}{1 + F \sin^2(\delta_T/2)}, \quad (7)$$

where

$$F = \frac{4\sqrt{R_1 R_s}}{(1 - \sqrt{R_1 R_s})^2}, \quad (8)$$

$$A = \frac{(1 - R_1)(R_2 R_3 - R_s)}{(1 - \sqrt{R_1 R_s})^2}, \quad (9)$$

and

$$B = \left| \frac{\sqrt{R_1} - \sqrt{R_s}}{1 - \sqrt{R_1 R_s}} \right|^2. \quad (10)$$

where the total round-trip phase is $\delta_T = (\delta_l + \bar{\delta}_s)$.

For the SIFP, in addition to a total round-trip phase δ_T , because of the use of R_s , both T and R also depend on the SI beam counterpropagating phase difference $\Delta\delta_s$. Similar to an active FP, an intensity-dependent nonlinear refractive material (NLM) can be used to modulate both R_s and $T(R)$ leading to an intensity-dependent cavity round-trip phase. The intensity-dependent round-trip phase, in turn, changes the cavity transmittance (reflectance) and results in multistable behavior of the equivalent FP cavity.

3. Active SIFP

A passive SI is a mechanically stable device. However, the use of intensity dependent nonlinear materials within the SIFP can cause the device to be optically multistable. Since different NLM placements in a SIFP can cause completely different transmission (reflection) effects, two different cases need to be considered. First, let the averaged SI phase $\bar{\delta}_s$ be fixed while the co-linear section phase δ_l vary. Because now the two SI counterpropagating waves traverse identical optical path, the SI phase difference is

zero or $\delta_{s1} = \delta_{s2}$. Thus, from eq. (4), $R_s = R_{sm}$ is fixed. This is similar to a co-linear FP case. Since the parameters R_2 and R_3 are not always equal to unity, in eq. (9), instead of the usual co-linear factor $(1 - R_s)$ the factor $(R_2 R_3 - R_s)$ is used. This replacement causes the SIFP transmission modulation to be lower than that of the corresponding FP. For an equal-ratio BS, the power transmission of eqs. (6) and (9) reduces to zero, while for a non-equal-ratio BS, transmission (reflection) expressions are similar to a conventional FP expressions. Thus, further discussions on its characteristics are omitted.

Next, we let the co-linear phase δ_l be fixed while we vary, with the pump intensity, the average phase $\bar{\delta}_s$. Because the two SI counterpropagating beams now traverse different optical paths, the SI phase difference is nonzero. As the phase difference $\Delta\bar{\delta}_s$ changes, R_s oscillates between its maximum value R_{sm} and zero. This configuration is equivalent to a co-linear FP with a single fixed and a variable reflectance mirror.

By inserting a cubic optical NLM in the SI loop, i.e. $n = n_0 + n_2 \langle E^2 \rangle$, and using an non-equal-ratio BS, an intensity-dependent non-zero phase difference $\Delta\bar{\delta}_s$ can be obtained [6,9]. Substituting in eqs. (6) and (7) the relations

$$\delta_T = \delta_{T0} + \Delta\bar{\delta}_s, \quad (11)$$

$$\begin{aligned} \Delta\bar{\delta}_s &= (2R_b - 1)\Delta\bar{\delta}_s \\ &= (2R_b - 1)(2\pi L/\lambda) n_2 \langle E^2 \rangle, \end{aligned} \quad (12)$$

where δ_{T0} , $\Delta\bar{\delta}_s$ and L are the initial total round-trip phase, the SI average phase change and the NLM cell length, respectively, results in an intensity-dependent SIFP transmission (reflection) curve. While with a FP tuning either the mirror reflectance or the initial resonator phase either affects the transmission modulation depth or translates the transmission curve, with a SIFP, because of different sine- and cosine-square function frequencies and initial phases in eqs. (4), (6) and (7),

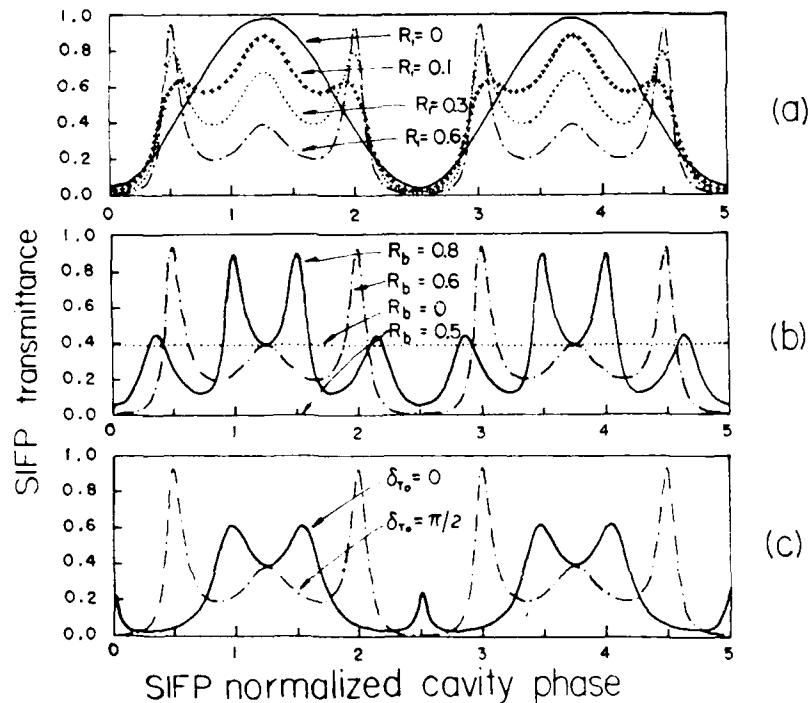


Fig. 2. Intensity transmission versus SIFP normalized resonator phase curves with adjustable parameters (a) input mirror reflectance R_1 , (b) BS reflectance R_b and (c) resonator initial phase δ_{T0} .

the corresponding tuning can result in radically different curves.

As examples, in figs. 2(a, b, c), three sets of the SIFP transmission versus input intensity curves, with different adjustable parameters R_1 , R_b and δ_{T0} are shown. In fig. 2(a) the effect of changing the front mirror reflectance R_1 is shown. For large R_1 , because in each period the peaks of the two off-center resonances are higher than the center resonance peak there are two dominant transmission regions. These multiple resonances are due to competitions between different sine- and cosine-square phase functions. Also, because of multiple-beam interference, the resonance bands are relatively narrow. As R_1 decreases, the resonance curve broadens and the center (side) transmission gradually increases (decreases). As R_1 approaches zero, in agreement with ref. [6] the off-center peaks recede yielding a single SI sinusoidal transmission curve. By varying the BS reflectance R_b , the curves of fig. 2(b) are obtained. Two extreme cases, when R_b is equal to 0.0 (1.0) and 0.5 (a equal-ratio BS), are of interest. In the former case, a constant bias ($1 - R_1$) $R_2 R_3$ curve is obtained indicating, that in this case, there is no interference effect. For a balanced (equal-ratio) BS, because on both sides of the NLM the SI incident intensities are identical, the SI phase difference is zero and, therefore, $R_s = R_{sm}$ is fixed. Although the intensity-dependent round-trip SI phase $\delta_s = \delta_{s1} = \delta_{s2}$ does vary, independent of δ_T , $(R_2 R_3 - R_s)$ is always equal to zero. Thus, except for a change in geometry, the above two special cases belong to a FP category. In fig. 2(c), the effect of varying the initial phase δ_{T0} is illustrated. Unlike a FP, where a change of the initial phase translates the transmission curve, here, a change in the initial phase, because it only affects the sine-square function, modifies the transmission curve.

To study resonator multistability phenomena, the SIFP transmission (reflection) curves of fig. 2 are utilized. Using the F-M method, taking transmission as an example, eq. (6) is interpreted as a solution of two simultaneous equations where the left hand side is a linear equation with a fixed slope as a function of input intensity while the right hand side is the nonlinear intensity-dependent transmission curve [10]. For a given slope, wherever multiple intersections of two curves can be found, multistable intensity transmissions may exist. When a pump beam illuminated

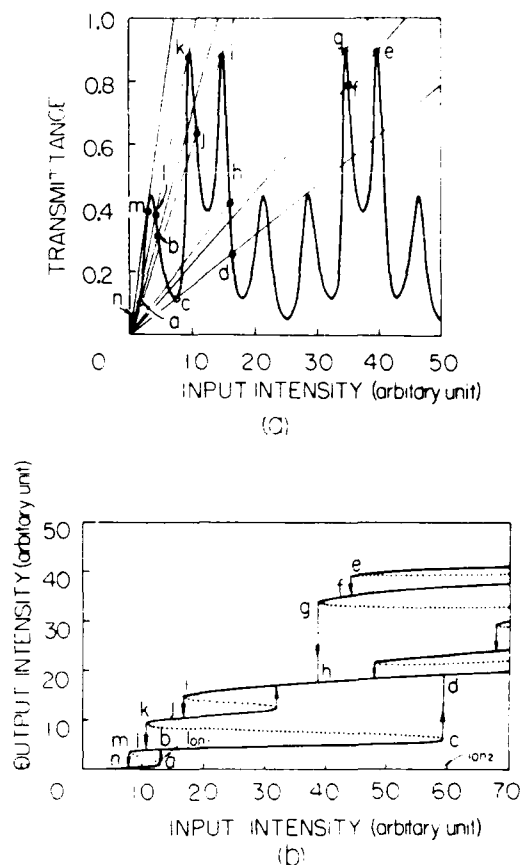


Fig. 3. (a) Typical SIFP transmission versus input intensity curve used for SIFP multistability analysis. Intersections of the straight line with the nonlinear transmission curve indicate multiple critical switching input and output intensities. (b) Multistable input and output SIFP intensity relations with corresponding critical switching points a, b through m, n .

co-linear section NLM is used, the corresponding multistable SIFP operators exhibits the same characteristics as does the usual co-linear FP. For this case both experimental and theoretical results are available [8,10,11]. When a pump beam illuminated SI section NLM is used, different multistable operation is expected. In fig. 3(a), indicated by the solid line, two cycles of a fig. 2(b) SIFP transmission curve is shown. The horizontal and vertical axes represent the input intensity and SIFP intensity transmittance, respectively. Initially, indicated by the intersection between

the transmission curve and the vertical coordinate, the SIFP is in a low transmission state. Increasing the input intensity decreases the slope of the straight line. The intersection then gradually moves up toward a point *a* where the straight line is tangent to the nonlinear curve. Correspondingly, the output intensity slowly increases. By increasing the input intensity (a further decrease in the straight line slope), a second intersection between the straight line and the nonlinear curve is sought. The closest intersection to *a* is the point *b* which is the next transmission state. Increasing the input intensity until the intersection reaches another tangential point *c* results in a slowly increasing output intensity. Past intersection *c*, another sudden jump brings the transmission state to a new intersection *d* resulting in another output intensity jump. Similarly, by continuously decreasing the output intensity the intensity-dependent transmission is forced to change from point *e* through *n*, and finally reaches a zero output intensity state. In fig. 3(b) the corresponding multistable intensity discontinuities are illustrated. The first (second) switch-on threshold input intensities are labeled as I_{on1} (I_{on2}), respectively. To obtain bistability, the input intensity should be below I_{on2} . For input intensities larger than I_{on2} , a number of output intensity levels can be probed. In principle, a SIFP and the nonlinear antiresonant ring interferometric switch proposed by ref. [6] have the same switching power requirement. In both devices, using an identical NLM, the switching power depends on the BS intensity transmittance (reflectance). The larger the imbalance between the values of the BS intensity transmittance and reflectance, the less the switching power is required. To lower the switching power requirement, materials with a large nonlinearity must be used. However, some large nonlinearity materials can have a severe carrier diffusion problem. For example, because the diffusion length is 60 μm in a InSb [12,13], nonlinear index modulation of a period less than this length will be washed out. To overcome the diffusion problem, the use of multiple-quantum-well semiconductor materials may be helpful [14]. It has been reported that semiconductor-doped glasses can produce large, fast and diffusion-free optical third order nonlinearities [15].

4. Applications

Finally, compared to a nonlinear FP, a nonlinear multistable SIFP is a flexible device with several additional adjustable parameters. The NLM may be placed in either one of two possible locations. With this device, in addition to the primary (cavity) input beam, using differently placed NLMs other external incident beams can also be used. As the overall inducing intensity exceeds the first switch-on threshold, both hysteretic and non-hysteretic bistabilities are obtainable. While a hysteretic bistability is suitable for optical memory and sequential logic operations, for asynchronous logic operation, a non-hysteretic bistability can be used to implement an optical limiter or a switch. The use of higher input intensities can cause SIFP multistable outputs. The tuning of initial phase, and other parameters such as BS reflectivity, can totally change the transmission curve. Properly choosing these parameters leads to different multistable operations. A SIFP multistability application is multistable optical switching. A combination of different SIFP multistable operation modes may help in the design of multiple-valued optical logic and arithmetic computing elements.

5. Summary

To summarize, nonlinear operations in a FP cavity consisting of a Sagnac interferometer and reflector are studied. A simple model that uses plane wave, and non-absorbing refractive nonlinearity assumptions is described. Different aspects of this active device are outlined. Applications in optical switching and computing are indicated.

Acknowledgement

Constructive comments by the referee are deeply appreciated. This work was supported in part by a grant from the Air Force Office of Scientific Research.

References

- [1] A.E. Siegman, IEEE J. Quantum Electron. QE-9 (1973) 247.
- [2] J.M. Buchert, D.K. Basa, C. Tzu and R.R. Alfano, J. Appl. Phys. 55 (1984) 683.
- [3] H. Vanherzeele, R. Torti and J.-C. Diels, Appl. Optics 23 (1984) 4182.
- [4] R. Trutna and A.E. Siegman, IEEE J. Quantum Electron. QE-13 (1977) 955.
- [5] Y. Li, G. Eichmann and R.R. Alfano, Appl. Optics 209 (1986).
- [6] K. Otsuka, Optics Lett. 8 (1983) 471.
- [7] G. Eichmann, Y. Li and R.R. Alfano, Opt. Eng. 25 (1986) 91.
- [8] D.A.B. Miller, IEEE J. Quantum Electron. QE-17 (1981) 306.
- [9] A.E. Kaplan and P. Meystre, Optics Comm. 40 (1982) 229.
- [10] F.S. Felber and J.M. Marburger, Appl. Phys. Lett. 28 (1976) 731.
- [11] S.D. Smith, A.C. Walker, B.S. Wherrett, F.A.P. Tooley, J.G.H. Mathew, M.R. Taghizadeh and I. Janossy, Appl. Optics 25 (1986) 1586.
- [12] H.A. MacKenzie, D.J. Hagan and H.A. Al-Attar, Optics Comm. 51 (1984) 352.
- [13] D.J. Hagan, H.A. MacKenzie, H.A. Al-Attar and W.J. Firth, Optics Lett. 10 (1985) 187.
- [14] D.A.B. Miller, D.S. Chemla, D.J. Eilenberger, P.V. Smith, A.C. Gossard and W.T. Tsang, Appl. Phys. Lett. 41 (1982) 679.
- [15] S.S. Yao, C. Karaguleff, A. Gabel, R. Fortenberry, C.T. Seaton and G.I. Stegeman, Appl. Phys. Lett. 46 (1985) 801.

HOUGH-TURN-TRANSFORM-BASED CIRCLE DETECTION USING AN ARRAY OF MULTIMODE OPTICAL FIBERS

Yao LI and George EICHMANN

Department of Electrical Engineering, City College of The City University of New York, New York, NY 10031, USA

Received 31 October 1986

The generation of an optical Hough transform (OHT) to detect a circle is proposed. The method is based on the use of a 2D multimode step-index optical fiber array. Both the position and radius of a circle can be detected. Some of the OHT performance parameters are also discussed.

1. Introduction

The Hough transform (HT) is an algorithm that transforms a particular curve to its corresponding parameter domain. The HT was originally introduced to detect straight lines in a two dimensional (2D) image [1]. The digital implementation of the HT has been extensively investigated [2-4]. Using the HT straight line detection concept, a generalized HT for the detection of other curves, such as circles, parabolas, ellipses, etc. have also been proposed and digitally implemented [5-8]. The generalized HT offers an effective means for digital image processing and pattern recognition. Recently, an optical HT (OHT) has been discussed. For OHT straight line detection, Eichmann and Dong [9] proposed a coherent space-invariant processor where the OHT is implemented by successively rotating the input image while translating the recording medium that records a 1D space-invariant transform. Another OHT implementation, proposed by Steier and Shori [10], uses a scanning dove prism and a linear detector array so that a faster processing speed can be achieved. In this letter, another space-variant OHT method for the detection of a circle using a 2D multimode step-index fiber array is proposed. This method is generally suitable for detecting all three parameters (i.e. radius and origin) of a circle.

2. Generalized Hough transform

In a plane cartesian coordinate system a straight line can be represented as

$$y = u_0 x + v_0, \quad (1)$$

where the two parameters u_0 and v_0 correspond to the straight line slope and its y coordinate intercept. Using the two parameters u and v as two new coordinates, the HT maps each cartesian point (x_0, y_0) to another straight line in (u, v) domain, i.e.

$$v = -x_0 u + y_0, \quad (2)$$

where $-x_0$ and y_0 serves as its new slope and the v axis intercept. If the input image is a straight line, it can always be decomposed into an infinite set of image points. In the HT domain, these infinite points are transformed into an infinite number of straight lines crossing at a common point (u_0, v_0) . Thus, for the straight line detection, the HT of a 2D binary image $f(x, y)$ can be expressed as

$$H(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(y - ux - v) dx dy, \quad (3)$$

where $\delta(\cdot)$ is the Dirac delta function. Similarly, to detect a circle

$$(x - u_0)^2 + (y - v_0)^2 = r_0^2, \quad (4)$$

where u_0 , v_0 and r_0 denote its origin and radius.

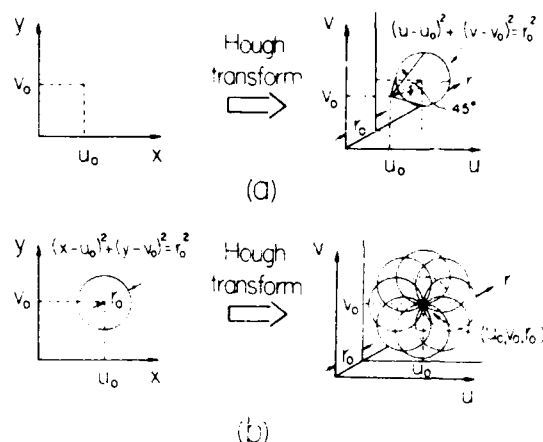


Fig. 1. The Hough transform for circular inputs. (a) A point (u_0, v_0) in the 2D Cartesian input (x, y) domain is transformed to a cone in the 3D (u, v, r) HT domain. The cone diverges at a 45° angle oriented toward the positive r direction. (b) A 2D input circle is transformed to a point (u_0, v_0, r_0) in the 3D HT domain. This point is formed by an intersection of infinite cones that are transformed from infinite input circle points.

respectively, the corresponding HT can be expressed as

$$H(u, v, r) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \times \delta[(x-u)^2 + (y-v)^2 - r^2] dx dy, \quad (5)$$

where again $f(x, y)$ is a binary input image and u, v and r are the three parametric HT coordinates. Thus, an input point in the (x, y) domain is transformed into a 3D cone that diverges along the positive r direction in the HT domain (see fig. 1(a)). When the input image contains a circle, an infinite set of cones, generated from the circle image points, will intersect at a common point (u_0, v_0, r_0) in the 3D HT coordinate system. In fig. 1(b), this transformation from a 2D circle into a point in the 3D HT domain is shown.

3. OHT implementation

For OHT circle detection, an optical space-variant transformation must be performed. In particular, a



Fig. 2. A schematic diagram showing the input and output angular relations for a multimode step-index fiber. For each angular input, a conical output illumination is formed.

point in the 2D Cartesian input domain (e.g. $x=u_0, y=v_0$) is to be transformed into a 3D conical surface whose tip starts at a point $u=u_0, v=v_0$ and $r=0$ and is flared at a 45° angle in the positive r direction (see fig. 1(a)). To generate such an impulse response, a multimode step index fiber can be used [11]. In fig. 2, a schematic multimode fiber light transmission diagram is shown. Light emanating from a set of displaced point sources is incident, at different entrance angles, into a multimode step-index fiber. Because of the existence of discrete fiber modes, all angular inputs can be assigned to separate and independent channels. At the output, light from each angularly multiplexed channel will be converted to a corresponding sheet of conical surface illumination. At an output screen, different size circles, corresponding to different angular inputs, are displayed. This angular multiplexing property of multimode step-index fibers has been utilized in many fiber optic applications, such as reduction of intermodal dispersion, enhancement of fiber transmission efficiency, geometrical transformations, etc. [11-13]. Here, the use of the angular multiplexing property of a multimode fiber is described for generalized OHT implementation.

For a fixed input coupling angle, the cone shaped output illumination is predetermined. Using a common input coupling angle to the 2D fiber array, an array of laterally displaced cone-shaped illumination is obtained. With an identical fiber packing sequence at both the input and output, this array can be used to implement a generalized OHT. In fig. 3, the proposed multimode step-index fiber-based OHT system is shown. The input illumination is a collimated beam of monochromatic light. To obtain the required input coupling angle, in front of the input plane, a computer-controlled acousto-optic (AO) deflector is used. The input image is an edge-enhanced 2D transparency that contains the circles to be detected. Next

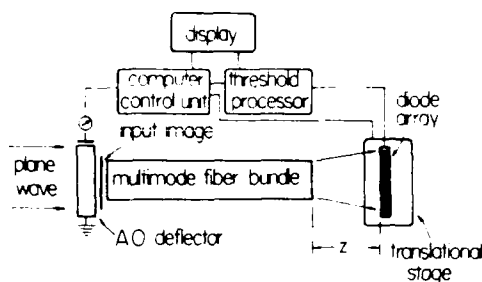


Fig. 3. A computer-controlled generalized OHT circle detection system. For obtaining different input angles, an AO deflector is used. Input image is located in front of the 2D multimode fiber array that performs the space-variant point to conical transformation. At the output, a 2D linear diode array mounted on a scannable translational stage is used. The signal is thresholded and displayed.

to the input transparency is a 2D multimode step-index fiber array that performs the generalized OHT. Since the output is in a 3D parameter domain, for detection, a 2D linear diode array mounted on a computer-controlled longitudinally scannable translation stage is used. To extract the correct output at the intersection point from the noise, the detected signal is thresholded. Finally, the three extracted parameters (u_0 , v_0 and r_0) are displayed.

There are two ways to generate the generalized OHT. With the first method, the input deflection angle ϕ_i is fixed while the longitudinal detector location (z) varies. In this case, a fixed output conical illumination is obtained for each input point. For input circles with identical radii centered at different origins, the output HT points are located on a plane which is at a particular longitudinal distance away from the fiber output plane. On the other hand, the HT points of concentric circles are distributed along the longitudinal direction. When the input contains circles with different radii centered at different origins, their generalized HT points are, in general, at different places in the 3D domain. Thus, a scan needs to be performed in the 3D parameter domain. When a 2D self-scannable diode array is used, only a longitudinal mechanical scan is to be performed. After thresholding, the peak obtained at a point (u_0 , v_0 , z_0) represents the position of the detected circle. (u_0 , v_0) and its radius $z_0 \tan \phi_i$, where $\tan \phi_i$ is a longitudinal scaling factor. With the second method, the angular multiplexing property of the fiber is utilized.

Here, instead of using a fixed input angle and longitudinal output scan, the input deflection angle is scanned while the output detector location z_i is fixed. In this case, in addition to the 2D diode array self-scan, the third dimension is the angle θ to be scanned by the input AO deflector. The three detected parameters are again u_0 , v_0 and ϕ_0 corresponding to the position of the input circle (u_0 , v_0) and its radius $z_i \tan \phi_0$. Because of the AO scan, the speed of detection is faster in this case.

4. OHT performance factors

Next, we discuss the resolution of the fiber-based OHT. The first factor to be considered is the fiber angular multiplexability, a capability that is important for the above-mentioned second detection method where the scanning of the input angle rather than the output distance is used. The guided modes of a step-index fiber can be characterized by the compound mode number m which is composed of an axial (l), and a radial mode (q) number, $m = l + 2q$. Corresponding to each compound mode m , a propagation angle ϕ_m can be assigned where

$$\phi_m = \lambda m / 4an, \quad (6)$$

and λ is the input wavelength in air, and a and n are the step-index fiber core radius and refractive index, respectively. From eq. (6), the angular spacing between two adjacent modes is

$$\Delta\phi = \phi_{m+1} - \phi_m = \lambda / 4an. \quad (7)$$

Assume that the input acceptance angle (inside the core) is ϕ_1 , then the number of angular channels is [13]

$$N = \phi_1 / \Delta\phi. \quad (8)$$

For example, for $\lambda = 0.8 \mu\text{m}$, $a = 100 \mu\text{m}$, $\phi_1 = 0.13 \text{ rad}$ and $n = 1.5$, in principle, a hundred angular channels are obtainable. This number allows the possibility of a smooth angular scan. In practice, however, because of mode coupling (crosstalk) the number of usable channels is limited. Due to non-uniformity of the refractive index, lack of core-cladding interface smoothness, etc., light in one angular channel will cross into other channels (crosstalk) resulting in a decrease in the number of multiplexable channels.

Also, due to the angular broadening of power distribution within each channel, the crosstalk increases with fiber length. Thus, for OHT implementation short (less than a meter) high quality fibers should be used.

Other factors to be considered are the OHT input image size and its resolution which are determined by the size of the 2D fiber array, by the fiber input acceptance angle, and by the available range of the output longitudinally scanning distances. For example, to detect a circle of 1.5 cm radius using fibers with overall core-cladding radius of 150 μm , a bundle of at least ten thousands fibers is needed. This number corresponds to an array packing density of about 1400 fibers/ cm^2 . This density sets the lower limit for circle detection. At this limit, circles with radii less than a few hundred microns can not illuminate sufficiently many fibers for correct angular transformation. This packing density also sets the upper limit for OHT position and radius resolutions, i.e. Δu_0 , Δv_0 and Δr_0 . There are distances below which two circles with either their origin or radius shifted can not be distinguished. The resolution of the position and radius also depends on the angular spread of each fiber mode, the longitudinal focusing distance as well as ray diffraction effects. Assuming a practical angular channel spread (including diffraction effect) and the required output longitudinal propagation distance be $\Delta\phi_p$ and z_0 , respectively, the position and radius resolutions can be found to be

$$\begin{aligned}\Delta r_0 &= (\Delta u_0^2 + \Delta v_0^2)^{1/2} = \Delta\phi_p z_0 \\ &= \Delta\phi_p r_0 / \tan \phi_i.\end{aligned}\quad (9)$$

For input circles with larger radii or for longitudinal focusing distance, the resolution of the output position decreases. Since the maximum input acceptance angle of the fiber is limited (to about 0.25 rad. with a glass fiber), detection of a circle of 1.0 cm radius requires a 5 cm longitudinal focusing distance. For circles with larger radii, the required increase in distance between the fiber output and detector plane may cause a diffraction problem. To shorten the longitudinal scanning distance, a fiber with a large input

acceptance angle, corresponding to a high fiber core refractive index, should to be used.

5. Summary

In this letter, a new method to perform a generalized OHT is proposed. The space-variant OHT is implemented with a 2D array of optical multimode step-index fibers. This method is suitable to detect both the origin and radius of a circle. Using a 2D fiber and a scannable detector array, all three parameters, i.e. the position and the radius of the input circle, can be detected. The operational principles of the setup have been described and some factors involving the multimode fiber performance were discussed.

Acknowledgement

This work was supported in part by a grant from the U.S. Air Force Office of Scientific Research.

References

- [1] P.V.C. Hough, U.S. Patent 3,069,654 (1962).
- [2] R.O. Duda and P.E. Hart, Commun. ACM 15 (1972) 11.
- [3] S.D. Shapiro and A. Iannino, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-1 (1979) 310.
- [4] W.K. Pratt, Digital image processing (Wiley-Interscience, New York, 1978).
- [5] C. Kimme, D. Ballard and J. Slasky, Commun. ACM 18 (1975) 120.
- [6] P.M. Merlin and D.J. Farber, IEEE Trans. Comput. 24 (1975) 96.
- [7] S. Tsuji and F. Matsumoto, IEEE Trans. Comput. 27 (1978) 777.
- [8] D.H. Ballard, Pattern Recognition 13 (1981) 111.
- [9] G. Eichmann and B.Z. Dong, Appl. Optics 22 (1983) 830.
- [10] W.H. Steier and R.K. Shon, Appl. Optics 25 (1986) 2734.
- [11] U. Levy, H. Kobrinsky and A.A. Friesem, IEEE J. Quantum Electron. QE-17 (1981) 2215.
- [12] G.J. Herskowitz, H. Kobrinsky and U. Levy, Laser Focus 19 (1983) 83.
- [13] J. Cederquist and A.M. Tai, Appl. Optics 23 (1984) 3099.